פתרון בעיית התאמת יתר ברשתות נוירונים עמוקות

ד"ר נוריאל ש. מור

נובמבר 2017

1. תקציר

בעבודה זו נסקרו טכניקות מרכזיות ועדכניות להפחתת התאמת יתר שהפכו כיום את רשתות נוירונים עמוקות לשיא הטכנולוגיה, בבעיות של זיהוי אובייקטים, זיהוי דיבור, וסיווג נתונים מגוונים בעלי יחס מאוד מורכב בין הקלט לפלט.
הטכניקות שנסקרו הן:

Unsupervised pre-training, Deep sparse rectifier neural networks & Dropout

טכניקות אלו הושוו, ונותחו בהרחבה, גם בהיבט התיאורטי.

Unsupervised pre-training פותחה ע"י הינטון ושות' ב 2007, היא אימון לא מונחה מקדים של הרשת לפני הBackprop. טכניקה זו נמצאה יעילה מאוד בהפחתת התאמת יתר ברשתות עמוקות בהשוואה לטכניקות אחרות שהיו קיימות עד לשנת 2007.

Deep sparse neural networks פותחה ע"י Bengio ושות' ב2011. ברשת זו נעשה שימוש ב Rectified linear unit (ReLU) כפונקציית אקטיבציה, במקום Sigmoid או Tanh. שימוש בפונקציית אקטיבציה זו נמצא יעיל יותר מ Unsupervised pre-training בהפחתת התאמת יתר, וייתר את הצורך ב Unsupervised pre-training.

Dropout טכניקה שפותחה ע"י הינטון ושות' ב2012. בטכניקה זו, לכל דוגמת קלט במהלך שלב האימון, מוחקים באופן זמני כל נוירון בהסתברות מסוימת, מבצעים Backprop, ומעדכנים את המשקולות. לאחר מכן, מחזירים את כל הנוירונים, וחוזרים על ההליך מחדש. טכניקה זו נמצאה מאוד יעילה בהפחתת התאמת יתר ומשולבת ברשתות עמוקות במערכות שיא הטכנולוגיה כדוגמת רכב ללא נהג.

במהלך ניתוח השיטות, והסקירה הנרחבת של הספרות, לשם מציאת הסברים תיאורטיים ליעילות של שלושת שיטות אלו, נמצא ש Sparse hidden units activation משותף לשלושת השיטות אלו. מרכיב זה מעניק יעילות לשלושת שיטות אלו בהפחתת התאמת יתר. למרות העובדה ש Unsupervised pre-training ו Dropout אינן מכוונות כלל ליצירת Sparsity.

בנוסף, בוצעה סקירה השוואתית של הרשתות:

.AlexNet, 2012; VGG Net, 2014; GoogleNet, 2014; & ResNet, 2015

ברשתות אלו שולבו הטכניקות שסוקרו בעבודה מסכמת זו. רשתות אלו הניבו ביצועים מיטביים ב ImageNet large scale visual recognition challenge. נמצא, שלעומק הרשת קיימת חשיבות קריטית בשיפור הביצועים.

על בסיס שתי המסקנות המרכזיות של עבודה מסכמת זו: 1. Sparse hidden units activation אלמנט מרכזי בהפחתה מוצלחת של התאמת יתר ברשתות עמוקות. 2. הצורך ברשת עמוקה מאוד לשיפור הביצועים במטלות מורכבות. הוצעה טכניקה חדשה מקורית להפחתת התאמת יתר, המיועדת לאפשר עיצוב רשתות מאוד עמוקות.

פתרון בעיית התאמת יתר ברשתות נוירונים עמוקות

2. מבוא

רשתות נוירונים עמוקות מכילות שכבות רבות לא ליניאריות, מה שגורם להן להיות מודלים מאוד אקספרסיביים שמסוגלים ללמוד יחסים מאוד מורכבים בין הקלט לפלט. בעיית התאמת יתר אינהרנטית לרשתות עמוקות מרובות פרמטרים. הרשת מתאימה באופן מושלם לדוגמאות האימון אך ללא יכולת הכללה לדוגמאות חדשות. במהלך ה 30 שנים האחרונות מאז פיתוח אלגוריתם הלמידה ברשתות רב שכבתיות (Backpropagation) הומצאו טכניקות רבות להפחתת בעיית התאמת יתר. כדוגמת:

.Weight decay, Weight sharing, Early stopping, Noise injection & Model averaging

אך טכניקות אלו לא היו יעילות מספיק להתמודדות עם התאמת יתר ברשתות עמוקות מרובות פרמטרים. תחום רשתות נוירונים עמוקות, היה בסטגנציה במשך עשרות שנים. עד שבשנת 2007 הינטון ושות' הציגו את הטכניקה, Unsupervised pre-training, שהפחיתה משמעותית את התאמת יתר ברשתות עמוקות והעניין המחקרי ברשתות עמוקות זכה לתחייה מחודשת. בשנת 2011 גילוי חדש לגבי היעילות של Deep sparse rectifier neural networks במניעת התאמת יתר, הזניק את תחום למידה עמוקה, וייתר את הצורך ב Unsupervised pre-training. שנה לאחר מכן, פותחה טכניקה נוספת הקרויה Dropout, ע"י הינטון ושות', שהפך את הלמידה ברשתות עמוקות לשיא הטכנולוגיה בבעיות של למידת מכונה ובינה מלאכותית. עד שכיום בשנת 2017 רשת ResNet, מניבה ביצועים טובים יותר מביצוע אנושי, בזיהוי תמונות. העבודה המסכמת כוללת 7 פרקים.

פרק 3 עוסק במרכיבים החישוביים הבסיסיים של רשתות נוירונים מלאכותיות. **פרק 4** עוסק בעקרונות של רשתות עמוקות. **פרק 5** מתאר את מקור בעיית התאמת יתר, וסוקר, משווה ומנתח, את שלושת הטכניקות המרכזיות לפתרון בעיית התאמת יתר, שהוזכרו לעיל:

Unsupervised pre-training, Deep sparse rectifier neural networks, & Dropout

כחלק ממסקנות ניתוח הספרות, ותוך דיון תיאורטי, הועלתה ההיפותזה, שהעיקרון הנוירו-חישובי במוח הביולוגי: Sparse coding, ובהקבלה ברשתות נוירונים: Sparse hidden units activation , הוא המרכיב המאחד את שלושת השיטות, והמעניק יעילות דווקא לשלושת השיטות האלו, בפתרון בעיית התאמת יתר. **בפרק 6** של העבודה מוצגים ביצועים של 4 רשתות קנוניות בזיהוי תמונות

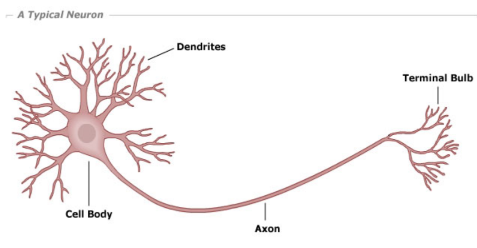
(AlexNet, 2012; VGG Net, 2014; GoogleNet, 2014; & ResNet, 2015) המשלבות את השיטות לפתרון בעיית התאמת יתר שנדונו בעבודה. **בפרק 7** של העבודה מוצעת טכניקה מקורית חדשה לפתרון בעיית התאמת יתר ברשתות עמוקות, המיועדת לאפשר עיצוב רשתות מאוד עמוקות. הטכניקה נשענת על מסקנות שהוסקו בעבודה זו, ועל עקרונות נוירו חישוביים נוספים.

3. רשתות נוירונים מלאכותיות

3.1 הנוירון המלאכותי הראשון

[6,7,10,11,32]

הנוירון המלאכותי הוצע לראשונה ע"י McCulloch & Pitts בשנת 1943. המוטיבציה של החוקרים הייתה לבנות מודל חישובי מתמטי של נוירון ביולוגי. המודל של McCulloch & Pitts פותח בהשראת ממצאי המחקרים בתחום מדעי המוח עד לאותה תקופה.



איור 1. מבנה נוירון ביולוגי
(Eberly college of science - human body: form & function)

נוירון ביולוגי מורכב מ 4 חלקים מרכזיים: ***דנדריטים, גוף התא, אקסון, ומחבר אקסון*** היוצר אזור מפגש עם נוירון נוסף. אזור מפגש זה קרוי ***סינפסה.***

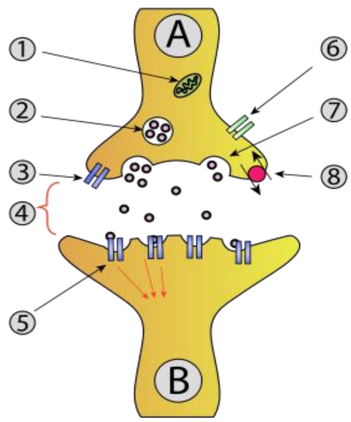
באופן כללי ביותר, ניתן לומר שהנוירון מקבל ***קלט*** מנוירונים רבים בחלק המכונה ***דנדריטים. בגוף התא*** מתבצע חישוב, המסכם את סך כל הקלט שהתקבל בדנדריטים. בעקבות החישוב אם סך כל הקלט גדול מערך סף, מתרחש פוטנציאל פעולה. ***באקסון*** מתבצעת הולכה של האות (פוטנציאל הפעולה). מהאקסון האות עובר לנוירון אחר דרך אזור המפגש הקרוי ***סינפסה.*** ראה איור 1.

על מנת להבין את הרציונל של McCulloch & Pitts בפרט, ושל רשתות נוירונים מלאכותיות בכלל, הנדונות בהרחבה בעבודה מסכמת זו. יש להבין מעט את הרקע הביוכימי של העברת האותות בין נוירונים, ואת המכניזם הביוכימי של למידה. מבנה מרכזי ביותר בלמידה ובהעברת אותות בין נוירונים הוא הסינפסה. מבנה זה מרכזי מאוד ברשתות נוירונים מלאכותיות.

3.1.1 סינפסה

[6,7,10,11,32]

כפי שצוין, אזור המפגש בין נוירונים מכונה סינפסה. אזור זה מוצג באיור 2.



איור 2. סינפסה
(נלקח מוויקיפדיה)

נתבונן באיור 2. הסינפסה היא בין קצה האקסון של נוירון A והדנדריטים של נוירון B. A הוא חלק מנוירון קדם-סינפטי המשדר לנוירון B, שהוא נוירון בתר-סינפטי.

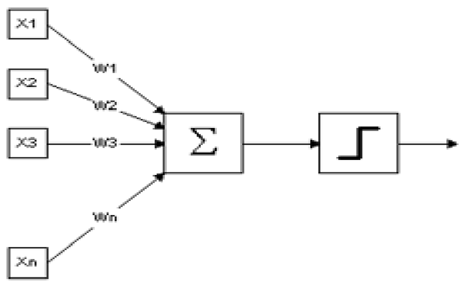
תיאור קצר של מעבר מידע מנוירון A לנוירון B.

האקסון של נוירון A מוליך אות חשמלי לאורך האקסון שלו. אות חשמלי זה מגיע לקצה האקסון של נוירון A. בקצה האקסון של נוירון A, כתוצאה מההגעה של האות החשמלי משתחררות אל המרווח הסינפטי מולקולות המכונות נוירוטרנסמיטרים. נוירוטרנסמיטרים אלו נקשרים לקולטנים בדנדריטים של נוירון B. כתוצאה מכך, נפתחות תעלות יונים בדנדריטים של נוירון B ומתח חשמלי מצטבר בגוף התא של נוירון B.
בגוף התא של נוירון B מתבצע החישוב הבא: אם המתח המצטבר גדול מערך סף, אזי יתרחש פוטנציאל פעולה שיעבור לאורך האקסון של נוירון B. אחרת, לא יועבר שום אות חשמלי לאורך נוירון B. לעקרון זה שאם הצטבר מתח חשמלי מעל לסף מסוים יתרחש פוטנציאל פעולה ויעבור אות, ואחרת, לא יתרחש פוטנציאל פעולה ולא יעבור אות, קוראים: ***All or none law.***

נוירון B מקבל קלט מנוירונים רבים בסינפסות רבות. חלק מהסינפסות הן ***סינפסות מעוררות,*** כלומר הנוירוטרנסמיטרים המשוחררים למרווח הסינפטי גורמים להעלאת המתח בגוף התא של נוירון B. חלק מהסינפסות הן ***סינפסות מעכבות,*** כלומר הנוירוטרנסמיטרים המשוחררים למרווח הסינפטי גורמים להורדת המתח בגוף התא של נוירון B. בגוף התא של נוירון B, מתבצע חישוב המסכם את המתח המצטבר - ואם ורק אם מתח זה גדול מערך סף, יתרחש פוטנציאל פעולה ויעבור אות חשמלי – אחרת, לא יועבר אות חשמלי. בנוסף, לכל סינפסה יש ***חוזק*** אחר. זהו

מושג מרכזי בהבנה של רשתות נוירונים מלאכותיות. יש סינפסות חזקות יותר, שמשפיעות יותר על המתח בגוף התא של נוירון B, ויש סינפסות חלשות יותר שמשפיעות פחות על המתח בגוף התא של נוירון B.

על סמך עקרונות אלו, McCulloch & Pitts [7,10,11,32] הציעו בשנת 1943 מודל מתמטי לפעילות הנוירון המוצג באיור 3.



איור 3. McCulloch–Pitts (MCP) Neuron

(נלקח מהבלוג מהי בינה מלאכותית? Kitty.in.th)

הנוירון המלאכותי מחשב את הסכום: $\sum w_i x_i$

כאשר הקלט x_i בינארי (0 או 1)

w_i משקולת, מספר ממשי.

אם הסכום $\sum w_i x_i$ המחושב בנוירון גדול מערך סף, הנוירון יחזיר פלט 1 , אחרת הנוירון יחזיר פלט 0 (פונקציית מדרגה).

ההקבלה של מודל מתמטי זה לנוירון המוחי:

x_i מסמן נוירון קדם-סינפטי. נוירון יכול להיות במצב 0 או 1 בהתאמה ל All or none law. 1 - נוירון שולח אות יש פוטנציאל פעולה. 0 – נוירון שקט, אין אות חשמלי, אין פוטנציאל פעולה.

w_i מסמן את חוזק וסוג הסינפסה. ערך גדול יותר מסמן סינפסה חזקה יותר עם השפעה גדולה יותר על הנוירון הבתר-סינפטי וההפך. בנוסף, הסימן + או – של w_i, מסמן את סוג הסינפסה מעוררת או מעכבת בהתאמה.

$\sum \boldsymbol{w_i x_i}$. הסכום שמחושב בגוף התא של הנוירון הבתר-סינפטי. אם סכום זה גדול מערך סף, יתרחש פוטנציאל פעולה, כלומר פלט=1 אחרת פלט=0. בהתאמה ל All or none law .

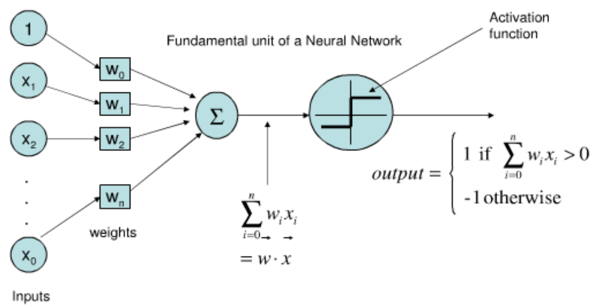
מבחינת הכוח החישובי של המודל של McCulloch & Pitts, לא קשה לראות שניתן לממש את כל הפונקציות הבוליאניות, באמצעות יחידות נוירונים אלו ע"י מימוש הפונקציות OR AND וNOT.

יש להבחין שמודל זה הוא תיאור מתמטי סטטי של פעילות נוירון בתר-סינפטי כתוצאה מפעילות נוירונים קדם-סינפטיים. כלומר המבנה וערכי המשקולות נקבעים פעם אחת ואינם משתנים. בסעיף הבא נראה כיצד 14 שנה לאחר הצגת המודל לפעילות נוירון של McCulloch & Pitts, פותח על בסיס המודל שלהם, אלגוריתם הלמידה הראשון ברשת נוירונים מלאכותית (פרספטרון) הכולל שינוי דינמי של ערכי המשקולות.

3.2 פרספטרון

[11,32]

בשנת 1957 הומצא ע"י פרנק רוזנבלט אלגוריתם הלמידה הראשון ברשת נוירונים מלאכותית. אלגוריתם זה קרוי פרספטרון. בדומה למודל של McCulloch & Pitts גם אלגוריתם למידה זה פותח בהשראת רשתות עצביות ביולוגיות. המחשבה המרכזית השולטת בחקר המוח היא שהביטוי הביוכימי של למידה מאופיין במוח באמצעות שינוי ביעילות **הסינפטית**, כלומר בחוזק של הסינפסות (Synaptic plasticity). כפי שראינו במודל של McCulloch & Pitts החוזק הסינפטי מיוצג באמצעות ערכי המשקולות w_i. מטרת אלגוריתם הלמידה של פרספטרון, להתאים את ערכי המשקולות, כך שלאחר ביצוע אלגוריתם הלמידה, הרשת תלמד לבצע קלסיפיקציה בינארית נכונה של הקלט.



איור 4. פרספטרון

(נלקח מ TeX.com: Diagram of a perceptron)

כפי שמתואר באיור 4. לאחר סיום שלב הלמידה, אם $\sum w_i x_i$ גדול מ 0 הפלט הוא 1 (הקלט מסווג לקבוצה 1) אחרת הפלט הוא 0 או 1- (הקלט מסווג לקבוצה 0). אם כן, הפרספטרון זהה לחלוטין, לאחר שלב אלגוריתם הלמידה, למודל של McCulloch & Pitts. נשאר להבין את אלגוריתם הלמידה של פרספטרון.

3.2.1 אלגוריתם הלמידה

[6,11,31,32]

ראשית יש לציין שסוג זה של למידה נקרא למידה מונחית (Supervised learning) מכיוון שמספקים לפרספטרון את הפלט הנכון עבור כל דוגמא, ועדכון המשקולות מתבצע בהתאם להבדל בין תשובת הפרספטרון לתשובה הנכונה. כלומר בשלב הלמידה, הפלט הנכון שהרשת הייתה אמורה להפיק מנחה אותנו כיצד לשנות את ערכי המשקולות.

אלגוריתם הלמידה או כלל הלמידה (שלב האימון):

אם הפלט נכון- לא משנים את המשקולות

אם הפלט בטעות 0 – הוסף את וקטור הקלט לווקטור המשקולות

אם הפלט בטעות 1 – החסר מווקטור המשקולות את וקטור הקלט

אלגוריתם זה מבטיח לנו שלאחר מספר סופי של איטרציות על כל דוגמאות האימון, נמצא וקטור של משקולות שיבצע סיווג נכון לכל דוגמאות האימון אם אכן יש וקטור משקולות כזה.

כאן המקום להגדיר שלושה מונחים שיצוינו בעקביות בספר זה.

הנתונים לרשת נוירונים מלאכותית מתחלקים לשלוש:

דוגמאות אימון. דוגמאות אימות. דוגמאות מבחן.

דוגמאות אימון (Training set) – דוגמאות שמשמשות לאימון הרשת ולעדכון המשקולות במהלך ביצוע אלגוריתם הלמידה.

דוגמאות אימות (Validation set) -דוגמאות שמשמשות להערכת טיב המודל (לאחר ביצוע אלגוריתם הלמידה) לשם בחינת הצורך בשינוי המודל. תוצאות שמתקבלות על דוגמאות האימות, עוזרות להחלטה לגבי הצורך בשינוי בפרמטרים של למידה או של ארכיטקטורה (מונחים אלו יובהרו בהמשך).

דוגמאות מבחן (Test set) – אלו דוגמאות חדשות לחלוטין שהרשת לא ראתה בשום שלב. הביצוע על דוגמאות המבחן למעשה מקביל לטיב הרשת בפתרון הבעיה, וליכולת ההכללה של הרשת לפתור את הבעיה בעולם האמיתי עבור כלל הדוגמאות.

3.3 מפרספטרון לרשת נוירונים כללית חד שכבתית

פונקציית האקטיבציה מגדירה את הפלט של הנוירון בהינתן הקלט ויש לה חלק מרכזי ביעילות הלמידה, כפי שנדון בהמשך.

כפי שראינו ב McCulloch & Pitts Neuron & Original Perceptron , פונקציית האקטיבציה היא פונקציית מדרגה (איור 3). אולם, פונקציות שונות ומגוונות יכולות להיות פונקציות אקטיבציה מתאימות. נזכיר כאן שלוש פונקציות אקטיבציה שכיחות: Sigmoid function, Hyperbolic tangent ו Rectified linear unit

3.3.1 אלגוריתם למידה - Gradient Descent

[6,11]

אלגוריתם הלמידה הכללי לרשת נוירונים חד שכבתית הוא למעשה יישום של Gradient descent על פונקציית המחיר שמעריכה עד כמה קרובה הרשת למטרה שאליה מכוון האימון.

3.3.2 פונקציית מחיר

[6,31,32]

כפי שראינו בפרספטרון המקורי אנו למעשה שואפים למצוא וקטור של משקולות שיבצע קלסיפיקציה נכונה לכל דוגמאות האימון. כלומר שההפרש בין הפלט של הרשת לפלט הנכון לכל דוגמאות האימון ישאף ל 0. נגדיר פונקציית מחיר (איור 5) שהיא למעשה מדד לשגיאה הכוללת של הרשת. מדד זה למעשה מבוסס על סכום ההפרשים בין הפלט של הרשת לפלט הנכון על כל דוגמאות האימון. מדד זה מאפשר לנו להעריך עד כמה קרובה הרשת למטרה שאליה מכוון האימון.

$$E = \frac{1}{2} \sum_{n \in training} (t^n - y^n)^2$$

איור 5. פונקציית מחיר -E.
t - פלט נכון y - פלט של הרשת

(Neural Networks for Machine Learning by University of Toronto, Geoffrey Hinton)

המטרה של אלגוריתם הלמידה הכללי היא למצוא בסופו של הליך וקטור של משקולות כך שמדד השגיאה שלנו - פונקציית המחיר - תהיה מינימלית. (נזכור שבפרספטרון המקורי המטרה הייתה למצוא וקטור משקולות כך שההפרש בין הפלט של הרשת לפלט הנכון לכל דוגמאות האימון יהיה 0)

ניתן לבצע זאת באמצעות Gradient descent בו אנו משנים את ערכי המשקולות כדי לרדת בהדרגה בערך פונקציית העלות. על מנת לרדת בערך פונקציית העלות אנו למעשה משתמשים בנגזרת של פונקציית העלות שמספקת לנו את הכיוון (שיפוע) שיש לשנות את המשקולות כדי לרדת בערך פונקציית העלות.

אם נגזור את פונקציית העלות שהגדרנו, נקבל שיש לשנות כל משקולת על פי הנוסחה (1). אציין שכלל זה ברשתות נוירונים חד שכבתיות מכונה כלל דלתא (Delta rule).

$$\Delta w_{ji} = \alpha(t_j - y_j)g'(h_j)x_i$$

נוסחה 1. Gradient Descent
t - פלט נכון. y - פלט של הרשת. x_i – ערך נוירון הקלט. **α – קבוע הלמידה:** מספר בתחום (0,1) הקובע את קצב הלמידה, קצב שינוי המשקולות בכל עדכון. $g(h_j)$ פונקציית האקטיבציה כאשר

$$h_j = \sum w_{ji} x_i$$

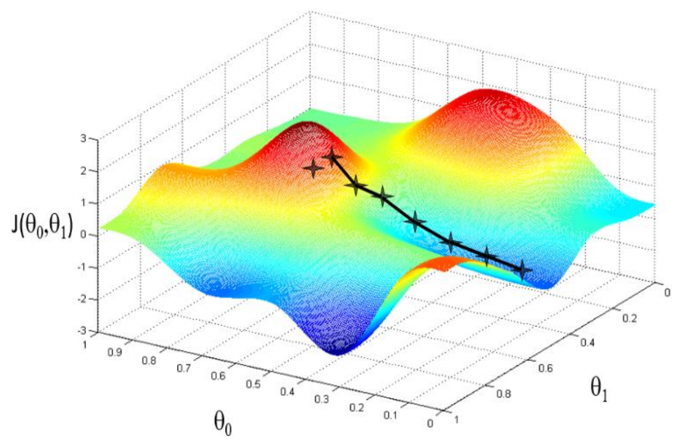
אם כן למעשה העיקרון של אלגוריתם הלמידה הכללי לרשת נוירונים חד שכבתית כללית:

1. חשב את הפלט של הרשת על דוגמת האימון

2. חשב את ההפרש בין הפלט שהתקבל לפלט הנכון שהיה צריך להתקבל

3. שנה כל משקולת על פי Gradient descent או כלל הדלתא

אציין שיש מספר מודיפיקציות לאלגוריתם הלמידה הנוגעות לגבי התזמון של שינוי ערכי המשקולות. ב Stochastic gradient descent משנים את המשקולות On line כלומר אחרי כל דוגמא. לעומת Batch gradient descent שבו משנים את המשקולות לאחר מעבר על כל דוגמאות האימון. Mini batch gradient descent הוא שילוב של שתי הגישות, ובו משנים את המשקולות לאחר מעבר על אחוז מסוים מסך הדוגמאות. לכל מודיפיקציה יש יתרונות וחסרונות בהקשר של מהירות ההתכנסות לווקטור משקולות אופטימלי, ותלויה בין היתר בגודל בסיס הנתונים. איור 6 ממחיש את הליך ה Gradient descent באופן גרפי. *הסימונים באיור:* J היא פונקציית המחיר ו Θ_0, Θ_1 הן המשקולות של הרשת.



איור 6. המחשה גרפית של Gradient Descent

(Machine Learning by Stanford University, Andrew Ng)

האלגוריתם מעדכן בכל שלב את המשקולות Θ_0, Θ_1 בהתאם לשיפוע כך שערך פונקציית המחיר (J) פוחתת עד להגעה למינימום המבוקש.

אם נחזור ונתבונן בכלל הלמידה של הפרספטרון המקורי, נראה שכלל הלמידה תואם לחלוטין ל Gradient descent במידה ונציב 1 במקום $g'(h_j)$. (כמובן שלא ניתן להשתמש ב Gradient descent כמו שהוא, בפרספטרון המקורי, כיוון שפונקציית המדרגות, פונקציית האקטיבציה, בפרספטרון המקורי איננה גזירה. אך ניתן לראות שכלל הלמידה של

הפרספטרון תואם ל Gradient descent במידה ומבצעים מודיפיקציה קטנה) לסיכום הראנו את אלגוריתם הלמידה הכללי (Gradient descent) לכל רשת נוירונים חד שכבתית עם פונקציית אקטיבציה מתאימה (גזירה באופן רציף). אלגוריתם הלמידה הכללי הצריך אותנו להגדיר פונקציית מחיר.

3.4 מגבלות של רשת נוירונים חד שכבתית והפתרון

[6,31,32,34]

בשנת 1957 כאשר פרנק רוזנבלט הציג את הפרספטרון הוא **טען שמחשב מבוסס על הפרספטרון יוכל ללכת, לדבר, לראות, לשכפל את עצמו ולהיות בעל מודעות.** אולם לאחר כ 10 שנים מהמצאת הפרספטרון, הוכח חד משמעית שהפרספטרון וכל רשת חד שכבתית אחרת מוגבלת מאוד מבחינת יכולת הקלסיפיקציה שלה. רשת חד שכבתית מסוגלת לבצע אך ורק הפרדה ליניארית. רשת חד שכבתית יכולה ללמוד לבצע קלסיפיקציה ל 2 קבוצות רק כאשר ניתן להפריד ליניארית (בקו ישר) את 2 הקבוצות. ממצא זה למעשה גרם להזנחת תחום של רשתות נוירונים למשך עשרות שנים. עד שבשנת 1986 שלושה חוקרים בשם

David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams [34]

הראו שאלגוריתם למידה ברשת רב שכבתית בשם Backpropagation algorithm מאפשר למידה ברשת רב שכבתית וכך ניתן להתגבר על המגבלה החישובית של פרספטרון ורשת נוירונים חד שכבתית.

אך לפני שנתחיל לדון ברשתות עמוקות, כדאי לדעת שישנו טריק מתמטי בשם Kernels המאפשר לרשתות חד שכבתיות לבצע הפרדה לא ליניארית. חשוב להבין ש Kernels לא מוסיף כוח חישובי או אקספרסיביות למודל, ובמקרים שבהם היחס בין הקלט לפלט מורכב מאוד כמו בבעיות מורכבות של אינטליגנציה מלאכותית (זיהוי אודיו, זיהוי אובייקטים וכו') Kernels לא יעניק לרשת חד שכבתית כוח חישובי לפתור בעיות אלו. זהו רק טריק מתמטי שמאפשר לרשת חד שכבתית לבצע הפרדה לא ליניארית, ויעיל במקרים מאוד מסוימים.

Kernels 3.4.1

[6,31]

Kernels הוא טריק מתמטי המאפשר לבצע הפרדה לא לינארית במקרים מסוימים, במודל שמבחינה אינהרנטית מוגבל אך ורק להפרדה ליניארית. כאשר יש יחס מאוד מורכב בין הקלט לפלט, Kernels לא יעזור. מכיוון שרשת חד שכבתית מספקת מודל לא מספיק אקספרסיבי.

מה שלמעשה עושים זה מניפולציה מתמטית על מאפייני הקלט כך שלאחר ביצוע המניפולציה עבור מאפייני הקלט החדשים הבעיה תהיה ניתנת להפרדה לינארית. לדוגמא אם ההפרדה היא מעגלית ויש שני מאפייני קלט, אז במקום מאפייני הקלט x_1, x_2 ניתן לעשות טרנספורמציה כך שמאפייני הקלט החדשים יהיו: $x_1, x_2, x_1x_2, x_1^2, x_2^2$ בהתאם לנוסחת המעגל, כעת ההפרדה לאחר הטרנספורמציה תהיה ליניארית.

וכעת ביתר פירוט:

ברשת חד שכבתית - עבור 2 מאפיינים (x_1, x_2) החישוב שאנו מבצעים:

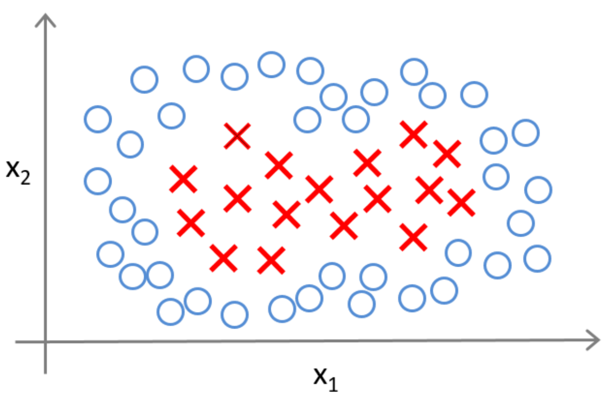
Predict 1 if

$w_0 + w_1x_1 + w_2x_2 > 0$

אולם ניתן לבצע מניפולציה מתמטית על הנוירונים של הקלט, כך שנוכל לבצע הפרדות לא ליניאריות. לדוגמא עבור הפרדה שהיא בצורה מעגלית/אליפטית כמו שמוצג באיור 7, נוכל לשנות את הנוירונים של הקלט (x_1, x_2) בצורה הבאה:

Predict 1 if

$w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2 > 0$



איור 7. הפרדה לא ליניארית

(Machine Learning by Stanford University, Andrew Ng)

אם אנחנו יודעים את הצורה של ההפרדה המבוקשת ניתן למצוא טרנספורמציה מתמטית מתאימה לנוירונים של הקלט על מנת שהרשת תוכל לבצע את הקלסיפיקציה הלא ליניארית, כפי שראינו לעיל. ישנה דרך כללית יותר לבחירת מאפייני קלט באמצעות Kernels.

הרעיון המרכזי הוא: בחר נקודות (גרעינים). **חשב** Similarity function בין מאפייני הקלט המקוריים לגרעינים שנבחרו. **התשובה** המתקבלת מה Similarity function יהיו **מאפייני הקלט החדשים** של הרשת החד שכבתית.

יש מספר Similarity functions אפשריות (Polynomial kernel, String kernel, Chi-square kernel, Histogram intersection kernel). השכיחה ביותר היא Gaussian kernel, המוצגת באיור 8.

Given x:

$$f_i = \text{similarity}(x, l^{(i)}) = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right)$$

איור 8. Gaussian Kernel

(Machine Learning by Stanford University, Andrew Ng)

האלגוריתם (*נלקח מ Machine Learning by Stanford university*):

Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$,
choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \ldots, l^{(m)} = x^{(m)}$.

Given example x:

$$f_1 = \text{similarity}(x, l^{(1)})$$
$$f_2 = \text{similarity}(x, l^{(2)})$$
$$\ldots$$

Predict $y = 1$ if $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

כאשר ה Similarity function יכולה להיות Gaussian כמובא באיור 8.

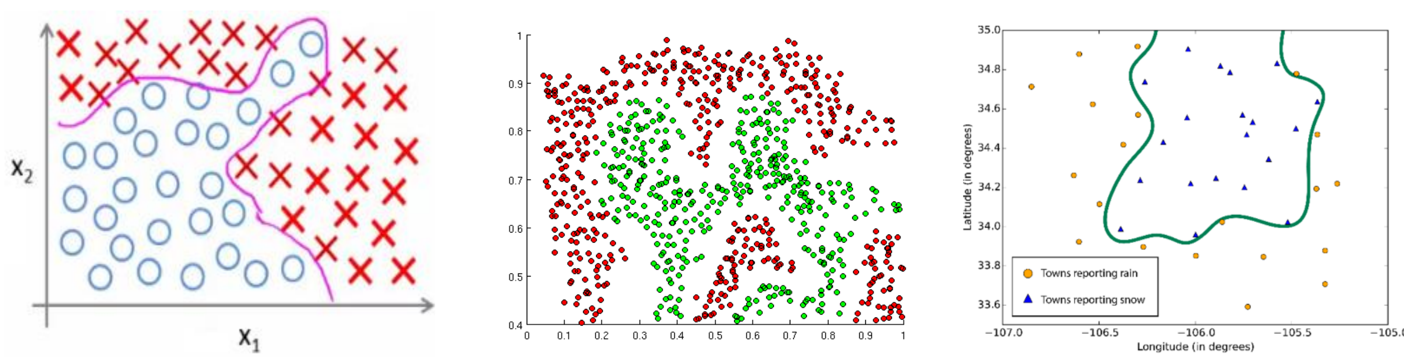
4. רשת נוירונים רב שכבתית (רשת עמוקה)

4.1 השראה מהמוח

[4,5,10,12,29,34]

הפתרון שהוצע בשנות ה80 למגבלה החישובית של רשת נוירונים חד שכבתית הוא הוספת שכבות חבויות. מההיבט התיאורטי/מתמטי הוספת שכבות, מוסיפה למודל כוח חישובי רב. למעשה, רשת עם מספיק שכבות ונוירונים יכולה להתאים (באמצעות משקולות מתאימות) באופן מושלם לכל יחס בין הפלט לקלט (בדוגמאות האימון), מורכב ככל שיהיה.

להבהרה ולהמחשה של יחס מורכב בין קלט לפלט אל מול בעיות שניתנות להפרדה ליניארית ראה איורים 9,10. בגלל המגבלה הגרפית, מוצגות בעיות קלסיפיקציה עם 2 מאפייני קלט (2 הצירים).

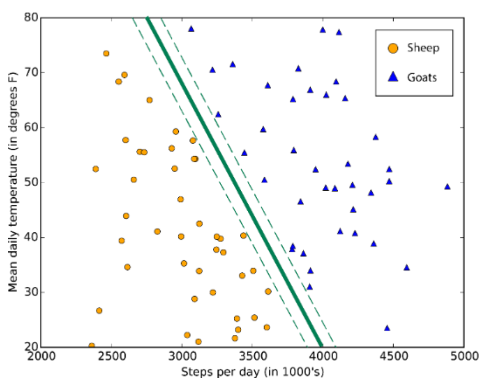


איור 9. לא ניתן להפרדה ליניארית

(How to choose algorithms for Microsoft Azure Machine Learning

Machine learning Non-linear classification Andrew NG- stanford.edu

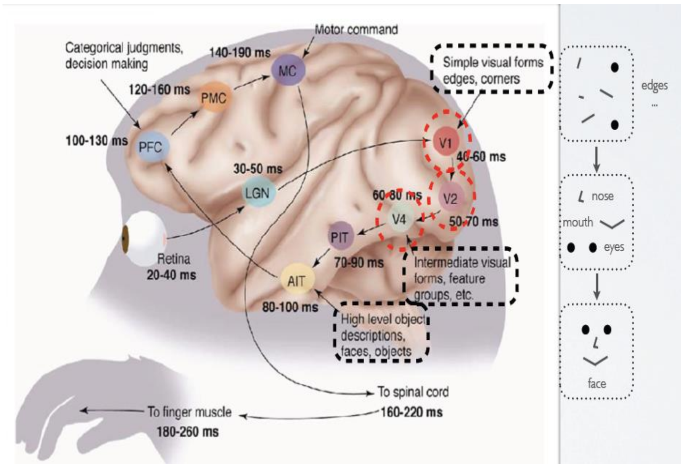
Neural networks - Overview and summary-holehouse.org)



איור 10. ניתן להפרדה ליניארית

(How to choose algorithms for Microsoft Azure Machine Learning)

המוטיבציה וההשראה לרשת נוירונים רב שכבתית כמו ההשראה לנוירון המלאכותי ולרשת חד שכבתית גם היא נובעת ממבנה המוח. המחשבה היא שהעיבוד במוח מתבצע על פני מספר שכבות כאשר כל שכבה שולחת את הפלט שלה לשכבה הבאה לעיבוד נוסף מתקדם יותר. הדוגמא הקלאסית להליך עיבוד רב שכבתי היא הליך זיהוי פנים במערכת הראייה במוח. כל שכבה במוח לומדת ייצוג מורכב יותר של הפרצוף, המורכב מצירופים פשוטים יותר המתקבלים משכבה קודמת. איור 11 מדגים הליך זה של זיהוי פנים המתבצע למעשה ברשת נוירונים רב שכבתית מוחית.



איור 11. זיהו פנים. רשת רב שכבתית מוחית

במוח, שכבה ראשונה V1 מגיבה לצורות פשוטות, קצוות, וקשתות. שכבה שנייה, V2 מגיבה לצרופים של צורות השכבה הראשונה, וכך הלאה עד שבשכבה האחרונה יש ייצוגים גבוהים של אובייקטים מורכבים, כדוגמת זיהוי פנים וכו'

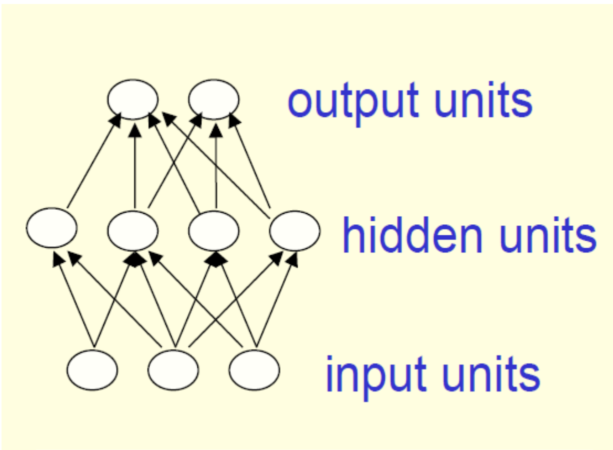
(Centre de Recherche Cerveau & Cognition, Simon Thorpe)

בדומה לרשת נוירונים חד שכבתית גם רשת נוירונים רב שכבתית למעשה מורכבת מ 3 מרכיבים מרכזיים. ארכיטקטורה, פונקציית אקטיבציה ואלגוריתם הלמידה.

4.2 ארכיטקטורה

[11,5,32,34,39]

הארכיטקטורה הכללית של רשת נוירונים רב שכבתית היא שכבת קלט, שכבות חבויות ושכבת פלט. כאשר יחידות נוירונים משכבה קודמת מחוברת במשקולת ליחידות הנוירונים בשכבה הבאה. איור 12 מציג ארכיטקטורה כללית של רשת נוירונים רב שכבתית בעלת שכבה חבויה אחת.



איור 12. ארכיטקטורה של רשת נוירונים רב שכבתית.

(Neural Networks for Machine Learning by University of Toronto, Geoffrey Hinton)

4.3 פונקציית אקטיבציה

[32]

בדומה לרשת רב שכבתית, לכל נוירון (מלבד לנוירונים בשכבת הקלט) יש פונקציית אקטיבציה המגדירה את הפלט של הנוירון בהינתן הקלט. ראה פרק פונקציות אקטיבציה ברשת חד שכבתית בנוגע לפרוט התכונות הרצויות של פונקציות אקטיבציה. הקלט לנוירון j בשכבה החבויה הראשונה דומה לרשת חד שכבתית והוא למעשה $h_j = \sum w_{ji} x_i$. כאשר x_i מייצג ערך של נוירון ביחידת קלט. w_{ji} מסמן משקולת מתאימה מנוירון קלט לנוירון j בשכבה חבויה ראשונה. הקלט לנוירון j בשכבות הבאות ממשיך באותו עקרון של סכום משוקלל ושווה ל $h_j = \sum w_{ji} h_i$ כאשר h_i הוא פלט של יחידת הנוירון בשכבה חבויה קודמת. w_{ji} מסמן משקולת מתאימה מנוירון i (משכבה קודמת) לנוירון j.

4.4 אלגוריתם למידה Backpropagation Algorithm
[34]

כפי שהוזכר, המרכיב המרכזי שמאפשר למעשה למידה ברשתות עם שכבות חבויות הוא אלגוריתם הלמידה Backpropagation algorithm.

ברשת חד שכבתית אנו יודעים לעדכן את המשקולות, ולהתקרב לווקטור משקולות מתאים שיפתור את הבעיה (במידה וניתנת להפרדה ליניארית) באופן מידי, מההפרש בין הפלט של הרשת לפלט הנכון (וביצוע Gradient descent על פני פונקציית המחיר). הבעיה ברשת רב שכבתית היא שאנו לא יכולים לדעת איך לעדכן את המשקולות שמחברות בין השכבות החבויות. כיוון שלא ניתן לדעת כיצד הנוירונים בשכבות החבויות צריכים להגיב. זהו הבדל עקרוני בין רשת רב שכבתית לרשת חד שכבתית. ברשת חד שכבתית המורכבת משכבת קלט ושכבת פלט אנו יודעים כיצד הנוירונים בשכבת הפלט צריכים להגיב. לעומת זאת ברשת רב שכבתית לא ניתן לדעת כיצד הנוירונים בשכבות החבויות צריכים להגיב. לכן כלל הלמידה ואלגוריתם הלמידה של רשת חד שכבתית לא יכול להתאים לרשת רב שכבתית.

בשנות ה 80 הראו ש Backpropagation algorithm מאפשר למידה ברשתות נוירונים רב שכבתיות. אנו לא יודעים כיצד הנוירונים בשכבות החבויות צריכים להגיב. אז במקום להשתמש בפעילות הרצויה של הנוירונים בשכבות החבויות נשתמש ב Error derivatives ובעברית ערכי השגיאה. ניתן לחשב את ערכי השגיאה לכל הנוירונים בשכבות החבויות ביעילות. ולעדכן את המשקולות בהתאמה. בכדי לחשב את ערכי השגיאה של כל נוירון בשכבה חבויה, אנו נעזרים בערכי השגיאה של הנוירונים מהשכבה הקודמת, כאשר סדר החישוב של ערכי השגיאה הוא משכבת הפלט לשכבת הקלט. בגלל ההליך של חלחול אחורה של ערכי השגיאה מהשכבות המתקדמות ברשת, האלגוריתם נקרא Back-propagation.

הפסקה הבאה מציגה את Backpropagation algorithm, את אופן חישוב גורם השגיאה ועדכון המשקולות.

Backpropagation Algorithm

נלקח מ Goswami, K., Park, Y., & Song, C. (2017)

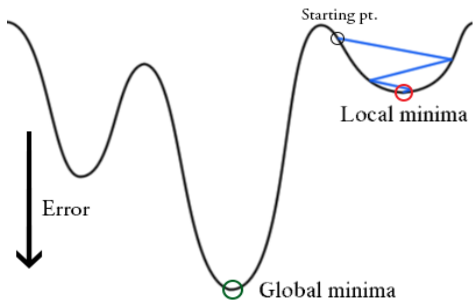
Algorithm 1 Backpropagation Algorithm

procedure TRAIN
- $X \leftarrow$ Training Data Set of size mxn
- $y \leftarrow$ Labels for records in X
- $w \leftarrow$ The weights for respective layers
- $l \leftarrow$ The number of layers in the neural network, 1...L
- $D_{ij}^{(l)} \leftarrow$ The error for all l,i,j
- $t_{ij}^{(l)} \leftarrow 0$. For all l,i,j
- $For \quad i = 1$ to m
  - $a^l \leftarrow feedforward(x^{(i)}, w)$
  - $d^l \leftarrow a(L) - y(i)$
  - $t_{ij}^{(l)} \leftarrow t_{ij}^{(l)} + a_j^{(l)} \cdot t_i^{l+1}$
- **if** $j \neq 0$ **then**
  - $D_{ij}^{(l)} \leftarrow \frac{1}{m} t_{ij}^{(l)} + \lambda w_{ij}^{(l)}$
- **else**
  - $D_{ij}^{(l)} \leftarrow \frac{1}{m} t_{ij}^{(l)}$
  - where $\frac{\partial}{\partial w_{ij}^{(l)}} J(w) = D_{ij}^{(l)}$

4.4.1 מינימום לוקלי או נקודות אוכף

[31,32]

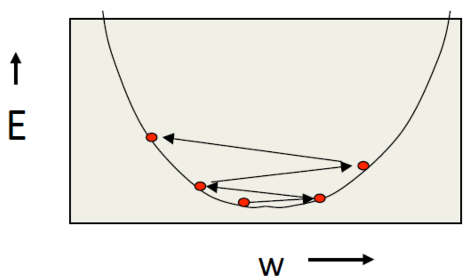
אחת הבעיות ברשתות רב שכבתיות שניתן לפתור בקלות יחסית היא קבלת ערכי משקולות סופיים שאינם מביאים את פונקציית המחיר למינימום גלובלי כנדרש. ייתכן ובמהלך ה Gradient descent נגיע למינימום לוקלי ונשאר שם, כל תזוזה לוקלית, שינוי במשקולות, תגרום לעלייה בערך פונקציית המחיר, מכיוון שאנו בתוך מינימום לוקלי, ולכן האלגוריתם יחליט לא לשנות את המשקולות. איור 13 מדגים את בעיית המינימום הלוקלי.



איור 13. מינימום לוקלי

(Training neural networks: backpropagation. Thinking & computing)

ישנם מספר פתרונות טובים לבעיה. ***אחד מהפתרונות*** זה התאמת קבוע הלמידה במהלך ה Gradient descent. קבוע למידה גבוה יכול לעזור בהימנעות ממינימום לוקלי כיוון שהשינוי במשקולות בכל צעד יהיה גדול מספיק בכדי לדלג מעל מינימום לוקלי. אך הבעיה בבחירת קבוע למידה גבוה הוא שהמשקולות יזוזו אחורה וקדימה לאורך עמק (Ravine) המינימום הגלובלי ולא יתכנסו למינימום הגלובלי עצמו (איור 14)



איור 14. קבוע למידה גבוה אין התכנסות למינימום

(Neural Networks for Machine Learning by University of Toronto, Geoffrey Hinton)

הפתרון הוא שינוי ערך קבוע הלמידה במהלך ה Gradient descent כאשר בתחילה יש לבחור בקבוע למידה גדול כך שנימנע מהיתקעות ממינימום לוקלי ורק בשלבים מאוחרים יותר (הגענו לערכי משקולות כך שפונקציית העלות בעמק המינימום הגלובלי) נקטין את קבוע הלמידה על מנת לאפשר התכנסות למינימום הגלובלי.

פתרון שני שמאוד נפוץ הוא שימוש בשיטת ***המומנטום***. למעשה המומנטום כמשמעותו הפיזיקלית היא, שיטה לבניית מהירות של שינוי המשקולות בכיוון שעקבי עם הגרדיאנט המיטבי. בשיטת המומנטום מעדכנים את המשקולות תוך לקיחה בחשבון לא רק של הגרדיאנט הנוכחי אלא גם של הגרדיאנטים הקודמים. באמצעות שיטה זו נמנעים ממינימום לוקלי באמצעות שינוי בצעדים גדולים יותר (מהיר יותר) של המשקולות בכיוון הגרדיאנט המבוקש וכך נמנעים

מינימום לוקלי. מומנטום נמוך מאט את ההתכנסות, ולא ישפיע על התכנסות למינימום לוקלי, אולם מומנטום גבוה מדי עלול לגרום לפספוס (קפיצה מעל) המינימום. לכן צריך גם כאן להוריד את ערך קבוע המומנטום בשלבים מתקדמים של הלמידה. יש להתאים את קבוע הלמידה ואת קבוע המומנטום לאורך הלמידה ולהבין שקבוע למידה וקבוע מומנטום גבוה יגרמו לבעיה בהתכנסות למינימום הגלובלי כאשר מתקרבים אליו.

ישנן עוד מספר טכניקות לפתרון בעיה זו של מינימום לוקלי. זו בעיה מינורית שניתן להתגבר עליה בקלות. קל למצוא מטריצת משקולות ברשת רב שכבתית כך שנקבל את המינימום הגלובלי לפונקציית המחיר. כלומר זו בעיה לא מסובכת להתאים את המשקולות כך שיתאימו באופן מושלם לדוגמאות האימון (הגעה למינימום גלובלי).

כאן המקום לציין הערה חשובה, בספרות משתמשים בביטוי מינימום לוקלי, כיוון שהאינטואיציה שלנו היא פונקציה עם 2 או 3 פרמטרים. אך למעשה מכיוון שמדובר כאן בפונקציות מרובות פרמטרים הסיכוי למינימום לוקלי כלומר שבכל הממדים הפונקציה תהיה קמורה **נמוך מאוד**. הנקודות הבעייתיות האלו ברשתות עמוקות הן למעשה נקודות אוכף ולא מינימום לוקלי.

ישנה בעיה אחרת לחלוטין ששמה ***התאמת יתר***. הגעה למינימום גלובלי והתאמה מושלמת לדוגמאות האימון עדיין לא אומר שהמודל שנבנה יהיה בעל ערך חיזוי עבור דוגמאות חדשות (דוגמאות מבחן). התאמת יתר הינה בעיה אינהרנטית לרשתות עמוקות ומסובכת מאוד לפתרון ותוצג בחלק הבא.

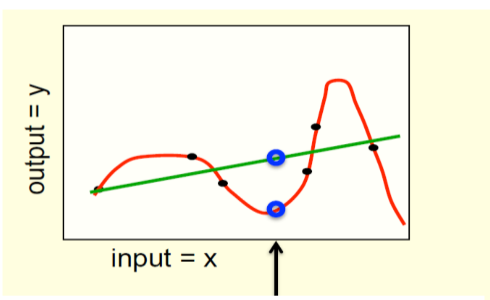
5. התאמת יתר ברשתות עמוקות

[3,12,13,18,31,32]

תיאורטית הוספת שכבות תמיד אמורה להוסיף כוח חישובי ולשפר את דיוק המודל. בניגוד לתיאוריה, נמצא שרשתות נוירונים רב שכבתיות לא מניבות תוצאות פרקטיות בזיהוי דפוסים (Pattern recognition). המכשול שמנע מרשתות רב שכבתיות להניב תוצאות פרקטיות, נעוץ בבעיה אינהרנטית לרשתות רב שכבתיות הקרויה התאמת יתר או Overfitting. הבעיה היא שכאשר יש לנו יחס מורכב בין פלט לקלט אנו חייבים להוסיף שכבות חבויות כדי שרשת הנוירונים תהיה אקספרסיבית מספיק כדי ללמוד את היחס המורכב בין הקלט לפלט. אך הוספת שכבות חבויות גורמת לרשת להיות כל כך אקספרסיבית כך שהיא מסוגלת ללמוד במדויק את היחס בין כל קלט לפלט בדוגמאות האימון ללא יכולת הכללה.

במילים אחרות הרשת מתאימה את עצמה יותר מדי (Over-fit) לדוגמאות האימון ללא יכולת הכללה לדוגמאות אחרות. ניתן לתאר במילים את בעיית התאמת יתר בכמה דרכים. דרך אחת היא שרשת עם שכבות חבויות רבות למעשה יכולה לשנן (Memorize) את היחס בין הקלט לפלט לכל דוגמא בדוגמאות האימון, ללא יכולת הכללה לדוגמאות חדשות. כלומר הרשת יכולה לפתור את בעיית הסיווג המוצגת בדוגמאות האימון בקלות באמצעות ***שינון*** היחס בין הקלט לפלט לכל דוגמה, ולא באמצעות ***למידה של היחס הנכון הכללי*** שיהיה נכון גם לדוגמאות חדשות שלא מוצגות בשלב האימון. דרך אחרת לנסח במילים את בעיית התאמת היתר היא שהרשת לומדת את היחס הנכון והמורכב שקיים רק בדוגמאות האימון כיוון שהיא לומדת גם את ***הרעש*** שנכון ספציפית רק בדוגמאות האימון ולא בדוגמאות חדשות.

איור 15 ממחיש את בעיית התאמת יתר ברשתות עמוקות. על איזה מודל, נסמוך שיתאר טוב יותר את היחס בין הקלט לפלט בעולם האמיתי? (דוגמאות חדשות). המודל המסובך מתאר בצורה מושלמת את היחס בדוגמאות האימון, אך מה עם יכולת ההכללה שלו? המודל הפשוט לא מתאים במדויק לדוגמאות האימון. אך באיזה מודל היינו בוחרים כדי לחזות את הפלט עבור דוגמא חדשה שהמודל לא ראה בשלב האימון? (החץ באיור)

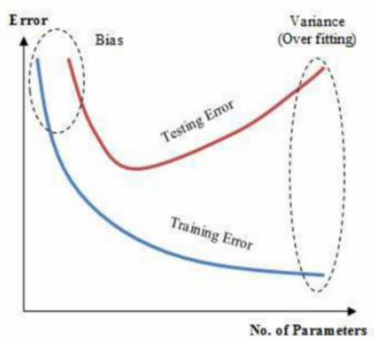


איור 15. בעיית התאמת יתר

(Neural Networks for Machine Learning by University of Toronto, Geoffrey Hinton)

הבעיה היא שהמודל מתאים באופן מושלם לדוגמאות האימון ומניב ביצועים עבור דוגמאות המבחן. נובע ממספר רב של שכבות חבויות ופרמטרים.

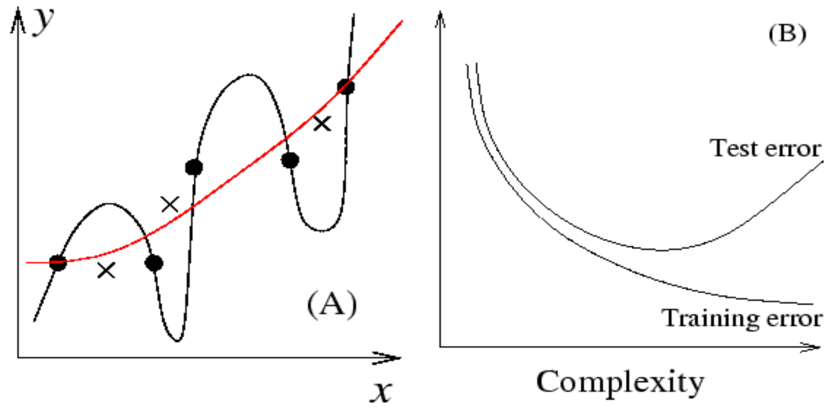
איור 16 ממחיש כיצד עלייה במורכבות הרשת גורמת מצד אחד להתאמה מושלמת יותר ויותר לדוגמאות האימון, כאשר מצד שני התוצאות על דוגמאות המבחן (דוגמאות חדשות) גרועות יותר ויותר כפונקציה של עלייה במורכבות הרשת (מנקודה מסוימת והילך). אנליזה זו קרויה Bias vs variance. ***High bias*** - משמע שהרשת לא מתאימה לדוגמאות האימון ובוודאי גם לא לדוגמאות המבחן - בד"כ אינדיקציה לכך שהמודל לא מספיק אקספרסיבי (יש להוסיף פרמטרים ושכבות) ואילו ***High variance*** – משמע שהרשת במצב התאמת יתר, ולא פרקטית עבוד דוגמאות חדשות (יש להוריד שכבות ופרמטרים). הבעיה כמובן היא שכאשר היחס בין הקלט לפלט מורכב מאוד, ***מוכרחים רשת עם מספיק שכבות חבויות ופרמטרים כדי למצוא מודל טוב עבור יחס מורכב***. אי אפשר לוותר על שכבות ופרמטרים כי אז המודל לא יהיה מוצלח אפילו על דוגמאות האימון. ואז מתעוררת בעיית התאמת יתר שאינהרנטית לרשתות עמוקות מרובות פרמטרים ושכבות.



איור 16. בעיית התאמת יתר כפונקציה של מורכבות הרשת

Kundu & Lad (2015)

איור 17, מציג את בעיית התאמת היתר מ2 פרספקטיבות. ***איור A*** מראה מודל מורכב (שחור) אל מול מודל פשוט יותר (אדום). ניתן לראות שלמרות שהמודל השחור מתאר באופן מושלם את היחס בין הקלט לפלט בדוגמאות האימון, אין למודל זה ערך פרקטי בחיזוי עבור דוגמאות חדשות. בעוד שהמודל האדום אינו מתאים במדויק לדוגמאות האימון אך בעל ערך לחיזוי הפלט לדוגמאות חדשות. ***איור B*** מראה את הטעות של המודל עבור דוגמאות האימון ועבור דוגמאות המבחן (דוגמאות חדשות) כפונקציה של מורכבות הרשת. ניתן לראות שככל שמורכבות הרשת עולה הטעות על דוגמאות האימון הולכת וקטנה אך מנקודה מסוימת של מורכבות, הטעות על דוגמאות חדשות הולכת וגדלה בגלל התאמת יתר.



איור 17. התאמת יתר.

(What are ways to prevent over fitting your training set data? Quora)

בעיית התאמת היתר מנעה מרשתות נוירונים עמוקות להניב תוצאות פרקטיות במשך יותר מ20 שנה. כלומר, רשתות עמוקות לא יכלו לייצר מודלים שיניבו חיזוי יעיל בעולם האמתי עבור דוגמאות חדשות (דוגמאות מבחן). לא נמצא פתרון טוב לבעיה משנות ה80 עד לשנת 2006. למרות שבשנים אלו (מ1980-2006) פותחו מספר טכניקות להפחתת התאמת יתר, טכניקות אלו, לא היו יעילות מספיק כך שרשתות נוירונים עמוקות יהיו פרקטיות . דוגמאות לטכניקות שפותחו הן:

Weight decay, Weight sharing, Early stopping, Noise injection & Model averaging

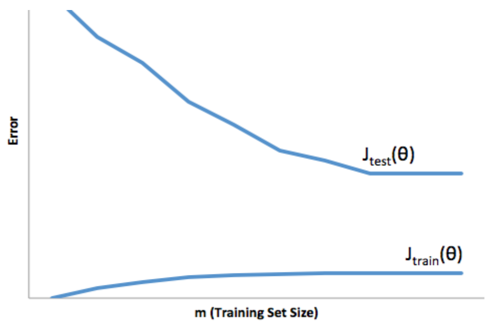
Unsupervised Pre-Training 5.1 להפחתת התאמת יתר

[5,12,13,14,19,20,21,33]

לפני שנדון ב Unsupervised pre-training צריך להבין את הרקע הטכנולוגי, ומדוע רק בשנת 2006 התאפשר לחוקרים למצוא טכניקות יעילות בפתרון התאמת יתר: ***מהירות המעבדים וזמינות של דוגמאות אימון רבות באינטרנט***.

1) מעבדים מהירים ויעילים יותר התאימו להרצה של אלגוריתם הלמידה ברשתות נוירונים עמוקות מרובות פרמטרים. אפשרו ביצוע ניסויים וקבלת תוצאות תוך פרק זמן סביר. כיום עם המעבדים המתקדמים ביותר, אימון רשת עמוקה מרובת שכבות ופרמטרים יכול לקחת מספר שבועות.

2) זמינות של דוגמאות אימון רבות בעקבות האינטרנט. בעיית התאמת יתר מחריפה כאשר אין מספיק דוגמאות אימון. מספר רב של דוגמאות אימון, גם כיום לאחר כל הטכניקות שפותחו, הוא הפתרון הטוב ביותר והיעיל ביותר לבעיית התאמת יתר. כשיש יותר ויותר דוגמאות אימון, לרשת יש הזדמנות ללמוד את היחס האמיתי בין הקלט לפלט שנמצא בעולם האמיתי ולא את הרעש שנכון באופן מקרי ליחס בין הקלט לפלט ***רק בדוגמאות האימון*** ולא בעולם האמיתי. איור 18 מדגים כיצד עלייה במספר דוגמאות האימון משפרת ביצועים על דוגמאות המבחן.



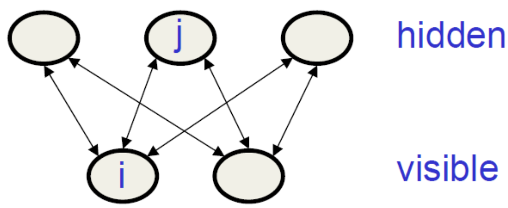
איור 18. השפעה של הוספת דוגמאות אימון על בעיית התאמת יתר.

(What are ways to prevent over fitting your training set data? Quora)

הטכניקה הראשונה שלמעשה החייתה את כל נושא למידה עמוקה ורשתות נוירונים עמוקות פותחה בשנת 2006 ושמה, Unsupervised pre-training

בשנת 1985 ג'פרי הינטון המציא אלגוריתם וארכיטקטורה ל Unsupervised learning הקרויה Boltzmann Machine (BM). זהו למעשה פיתוח של Hopfield Network. האנרגיה הכללית של BM זהה לאנרגיה הכללית של Hopfield Network. ב Hopfield Network שואפים להגיע למינימום אנרגיה, וב BM למצב שנקרא Thermal equilibrium. האלגוריתם של BM אינו יעיל, $O(k^n)$, ואינו שימושי.

Restricted Boltzmann Machine (RBM) בעל אותן עקרונות כמו BM רק עם ארכיטקטורה מוגבלת. יש משקולת רק בין יחידות משתי השכבות סמוכות ואין קשר בין יחידות באותה השכבה. איור 19 מציג דוגמא ל RBM.



איור 19. RBM

(Neural Networks for Machine Learning by University of Toronto,Geoffrey Hinton)

גם האלגוריתם של RBM אינו יעיל ואינו שימושי. אלא שלקראת 2006, הינטון וצוותו, מצאו קיצור דרך לאלגוריתם המקורי, שאפשר לבצע Generative unsupervised learning ב RBM באופן מהיר. לאלגוריתם הזה קוראים Contrastive divergence. בנוסף, בשנת 2006 הינטון וצוותו, מצאו דרך לחבר RBM אחד על גבי השני ולאמן את כל ה RMBs בצורה מהירה (כך שמתקבל מודל עמוק אחד). האימון המהיר של RMBs אחד על השני מתבצע באמצעות Greedy wise layer learning, אימון של שכבה אחת אחרת בכל פעם. הם קראו למבנה של RBM אחד על השני Deep Belief Net (DBN) .

לסיכום עד כאן, הינטון ושותפיו בשנת 2006 מצאו דרך לבצע Generative unsupervised learning ברשת עמוקה באופן מהיר ויעיל יחסית.

ובהקשר שלנו:

הם מצאו שביצוע Generative unsupervised training **<u>לפני ה</u>** Backpropagation algorithm מפחית את בעיית התאמת יתר!

ה Generative unsupervised training מניב ערכי משקולות. ועם ערכי המשקולות האלו מבצעים את Backpropagation algorithm הרגיל והמוכר, על מנת לבצע Fine tuning לערכי המשקולות ולשפר את דיוק הקלסיפיקציה. טכניקה זו נקראת בספרות **Generative unsupervised <u>pre</u>-training**, מכיוון שהיא מתבצעת לפני ה Discriminative backpropagation algorithm.

הפרטים של האלגוריתם המקורי של הליך הלמידה המתבצעת ב RBM והפרטים לגבי אופן ביצוע קיצור הדרך הנקרא Contrastive divergence יחד עם Greedy wise layer learning והסיבות מדוע קיצור הדרך מוצלח, הם עבודה בפני עצמה. ניתן לקרא ולהתרשם מפרטים אלו בבהירות ב[19,20,21]. באופן עקרוני האימון המסורתי של RBM ועדכון המשקולות הצריך מעבר סדרתי ארוך מאוד, בין היחידות הנראות (visible units) והיחידות החבויות. בשנת 2006 נמצא שמעבר אחד בין היחידות הנראות והיחידות החבויות ועדכון המשקולות על פי מעבר זה מייצר מודל טוב מספיק של הנתונים.

5.1.1 ניתוח השוואתי, בין ביצועי רשתות עם ובלי Unsupervised Pre-Training [19,20].

המאמר הקנוני שגרם לתחייה מחודשת בעניין המחקרי ברשתות עמוקות התפרסם ב2006 ע"י הינטון ושותפיו [19]. מאמר זה מבצע השוואה בין ביצועי רשת עם Unsupervised generative learning לרשתות אחרות ולשיטות אחרות של למידת מכונה. חשוב לזכור שעד אז לא היה ניתן לאמן ביעילות רשתות נוירונים עמוקות בגלל בעיית התאמת יתר. השיטות המוצלחות ביותר עד אז היו גרסאות של Support Vector Machines. הרשת המוצלחת ביותר עד אז שהתגברה על התאמת יתר, הייתה LeNet שהומצאה ב 1998, הסיבות ליעילות שלה ידונו מאוחר יותר בפרק מס' , אציין שעד היום משתמשים בה במערכות בנקאיות לזיהוי כתב וחתימות בהפקדות צ'קים.

מטרת הניסוי: השוואת השיטה החדשה ל Unsupervised generative learning (Contrastive divergence) מול אלגוריתמי למידה אחרים במשימת זיהוי ספרות שנכתבו בכתב יד. בסיס הנתונים היה MNIST שהוא בסיס נתונים מאוד מוכר וגדול של ספרות שנכתבו בכתב יד. בסיס הנתונים מכיל 60,000 דוגמאות אימון ו 10,000 דוגמאות מבחן. ניתן להשוות את ביצועי הרשת של הינטון ושותפיו, עם סוגים שונים של רשתות ושיטות, כיוון שבסיס נתונים זה מאוד שכיח בהרבה עבודות של למידת מכונה וPattern recognition.

תוצאות: ניתן לראות מאיור 20 שהרשת עם Generative learning של הינטון ושותפיו, הניבה רק 1.25 אחוז טעות על 10,000 דוגמאות המבחן. לרשת של הינטון ושות' הייתה 3 שכבות חבויות. בעוד שרשת עם 2 שכבות חבויות שאלגוריתם הלמידה היה Backpropagation עם Weight decay כמתודה להפחתת התאמת יתר, הניבה 1.51 אחוז טעות. רשת עם 2 שכבות חבויות עם Backpropagation ועצירה מוקדמת כמתודה להפחתת התאמת יתר, הניבה 1.53 אחוז טעות.

סיכום: במשימה של Pattern recognition, Generative learning של הינטון ושות' מוצלחת יותר מ Backpropagation בתוספת שיטות להפחתת התאמת יתר (Weight decay & Early stopping) זוהי תחילתה של נקודת המפנה.

יש לשים לב, למספר השכבות החבויות של כל רשת. ה Generative learning אפשרה להוסיף שכבות חבויות. כלומר להוסיף כוח חישובי ואקספרסיביות תוך כדי התמודדות טובה יותר עם התאמת יתר שנובעת מהוספה זו. ללא

Generative learning הרשתות הטבות ביותר הכילו מקסימום 2 שכבות חבויות. זה מרכיב שצריך לשים לב אליו, שמבצעים ניתוח השוואתי בין רשתות. במידה ומוסיפים שכבות חבויות ומצליחים לשפר את ביצועי הרשת בעקבות הוספת שכבות חבויות, זה אינדיקציה לטיפול טוב בהתאמת יתר.

התאמת יתר מתעוררת כאשר מוסיפים שכבות חבויות. למרות שבאופן תיאורטי הוספת שכבות חבויות אמורה תמיד לשפר את המודל. ניתן לראות שהשיטות ללא Generative learning היו עם 2 שכבות חבויות, כלומר השיטות לא אפשרו הוספה של שכבות בגלל התאמת יתר. אולם, הרשת של הינטון ושות' הייתה עם 3 שכבות חבויות כלומר השיטה שלהם אפשרה להוסיף שכבה חבויה, ולשפר המודל בעקבות הוספת השכבה. מכיוון שהשיטה שלהם מתמודדת ביעילות עם התאמת יתר.

במאמר זה שהובא לעיל, מ 2006 הינטון ושותפיו הציגו את Generative learning , מול אלגוריתמים אחרים. ה Generative learning במאמר קנוני זה לא הגיע כ Pre-training. רק מאוחר יותר הינטון פיתח את הרעיון של ביצוע Unsupervised pre-training ולאחר מכן Fine tuning באמצעות Backpropagation, כפי שמובא בפסקה הבאה.

איור 20 מציג את ההשוואה שבוצעה ע"י הינטון ושותפיו ב 2006.

Table 1: Error rates of Various Learning Algorithms on the MNIST Digit Recognition Task.

Version of MNIST Task	Learning Algorithm	Test Error %
Permutation invariant	Our generative model: 784 → 500 → 500 ↔ 2000 ↔ 10	1.25
Permutation invariant	Support vector machine: degree 9 polynomial kernel	1.4
Permutation invariant	Backprop: 784 → 500 → 300 → 10 cross-entropy and weight-decay	1.51
Permutation invariant	Backprop: 784 → 800 → 10 cross-entropy and early stopping	1.53
Permutation invariant	Backprop: 784 → 500 → 150 → 10 squared error and on-line updates	2.95
Permutation invariant	Nearest neighbor: all 60,000 examples and L3 norm	2.8
Permutation invariant	Nearest neighbor: all 60,000 examples and L2 norm	3.1
Permutation invariant	Nearest neighbor: 20,000 examples and L3 norm	4.0
Permutation invariant	Nearest neighbor: 20,000 examples and L2 norm	4.4
Unpermuted images; extra data from elastic deformations	Backprop: cross-entropy and early-stopping convolutional neural net	0.4
Unpermuted de-skewed images; extra data from 2 pixel translations	Virtual SVM: degree 9 polynomial kernel	0.56
Unpermuted images	Shape-context features: hand-coded matching	0.63
Unpermuted images; extra data from affine transformations	Backprop in LeNet5: convolutional neural net	0.8
Unpermuted images	Backprop in LeNet5: convolutional neural net	0.95

איור 20. ניתוח השוואתי Unsupervised Generative Learning אל מול שיטות אחרות

(Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.)

מאמר קנוני שהתפרסם ב2007 , ע"י ג'פרי הינטון [20] , הראה ש Unsupervised pre-training ולאחריו הביצוע הרגיל והמוכר של Discriminative backpropagation algorithm, מניב תוצאות טובות בהרבה, מרשתות ללא

Unsupervised pre-training. איור 21, מציג את ההשוואה. גם עבודה זו בוצעה על MNIST, והמשימה היא זיהוי ספרות שנכתבו בכתב יד. 60,000 דוגמאות אימון, ו 10,000 דוגמאות מבחן.

pre-trained network	backprop training set size	train epochs	train cost per 100	train errs	valid. cost per 100	valid. errs	test cost per 100	test errs
Neta	50,000	33	0.12	1	6.49	129	6.22	122
Netb	50,000	56	0.04	0	7.81	118	6.21	116
Netc	50,000	63	0.03	0	8.12	118	6.73	124
Combined							5.75	110
Neta	60,000	33+16	<0.12	1			5.81	113
Netb	60,000	56+28	<0.04	0			5.90	106
Netc	60,000	63+31	<0.03	0			5.93	118
Combined							5.40	106
not pre-pretrained	60,000	119	<0.063	0			18.43	227

Table 1: Neta, Netb, and Netc were greedily pretrained on different, unlabeled, subsets of the training data that were obtained by removing disjoint validation sets of 10,000 images. After pretraining, they were trained on those same subsets using backpropagation. Then the training was continued on the full training set until the cross-entropy error reached the criterion explained in the text.

איור 21. ניתוח השוואתי עם ובלי Unsupervised Pre-Training

(Hinton, G. E. (2007). To recognize shapes, first learn to generate images. Progress in brain research, 165, 535-547.)

ניתן להתרשם שרשתות שעברו Unsupervised pre-training ולאחר מכן Backpropagation הניבו כ1.1 אחוז טעות ורשתות עם Backpropagation ללא הליך מקדים הניבו 2.27 אחוז טעות. כלומר רשתות עם Unsupervised pre-training ולאחריו Backpropagation מניבות את התוצאות הטובות ביותר בשנת 2007.

מכיוון שהעבודה מ 2007 התבצעה על אותו בסיס נתונים MNIST כמו הפרסום הראשון מ 2006. נוכל אנו כאן לבצע עצמאית אנליזה נוספת (איור 20 ואיור 21). וניווכח שרשת עם Unsupervised generative pre-training ולאחריה Backpropagation מניבה 1.1 אחוז טעות ויעילה יותר מרשת עם Unsupervised generative learning לבד ללא Backpropagation המניבה 1.25 אחוז טעות.

5.1.2 מדוע Unsupervised Pre-Training מפחית התאמת יתר?
[13,20,27,32]

1. ***מנקודת ראות של אופטימיזציה.*** המשקולות שאנחנו מתחילים איתם את ה Backpropagation algorithm הם לא רנדומליים אלא אנחנו מתחילים עם ערכי משקולות שיכולים מאוד לעזור ב Discriminative task שמתבצע ב Backpropagation. כלומר נקודת הפתיחה היא מאוד טובה עבור ה Backpropagation . הגרדיאנטים הראשוניים לפני התחלת הBackpropagation כבר טובים ויש לבצע רק חיפוש לוקלי מנקודת התחלה טובה.

2. ***מנקודת ראות של התאמת יתר***. הקלט לבדו מכיל הרבה מאוד מידע, יותר מ ה Label (Label = הפלט המתאים לכל קלט), שחשוב ל Discriminative task. כשמבצעים Unsupervised pre-training לא מבזבזים את המידע החשוב הזה שהקלט לבדו מכיל. ערכי המשקולות הסופיים נקבעים בעיקר מהמידע שהקלט מכיל. כפי שראינו במאמר הקנוני הראשון של הינטון ושות', Generative learning בלבד מצליח לספק ערכי משקולות טובים יותר מ Backpropagation שמשתמש רק ב Label, ולא במידע שמכיל הקלט. כפי שראינו אחר כך, במאמר הקנוני השני של הינטון, ביצוע של Backpropagation לאחר ה Generative pre-training משפר עוד יותר את ביצועי הרשת. אנו משתמשים במידע המאוד יקר ש ה Label מספק לנו רק בסוף, ב Fine tuning. כלומר ה-Unsupervised pre-training מגלה את כל ה Features החשובים (מגולם בערכי המשקולות) מהמידע שבקלט. וה Fine tuning משנה מעט את Features רק למקם נכון את ה Decision boundary, כך שנבצע טוב יותר את הקלסיפיקציה.

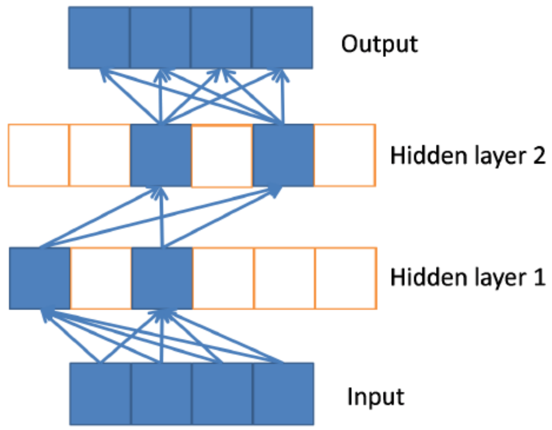
5.1.3 Unsupervised Pre-Training & Sparsity?
[8,27,28,40]

נקודה תיאורטית מעניינת שלא נדונה בהרחבה בספרות המקצועית בהקשר להשפעה של Unsupervised pre-training, היא ש Unsupervised pre-training גורם למודל דליל, כלומר, לדלילות בפעילות של הנוירונים בשכבות החבויות. לפני שארחיב על ההשפעה של Unsupervised pre-training על דלילות המודל, אסביר בקצרה מה הכוונה במודל דליל (Sparsity).

הסבר כללי בנוגע לSparsity

עבור כל דוגמת קלט, רק תת קבוצה קטנה ספציפית מסך כל הנוירונים פעילה, בתגובה לאותו הקלט (עבור דוגמאות קלט שונות תת קבוצה שונה פעילה)

איור 22 ממחיש Sparse hidden units activation



איור 22. דוגמא לכוונה במודל דליל (Sparsity). רק חלק קטן מאוד מהנוירונים בשכבות החבויות פעיל בתגובה לכל קלט.

(Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics)

Sparse coding הוא מושג מרכזי ב Theoretical neuroscience ובמדעי המוח. זו למעשה הדרך שבה המוח הביולוגי מקודד אינפורמציה, ומה שמעניק למוח יעילות בביצוע קלסיפיקציות וקיבולת מוגברת. נדון בהרחבה ב Sparse coding ו Model sparsity בהמשך.

Unsupervised Pre-Training Encourages Sparseness
[27]

Li, Luo, Yang, and Yuan [27] מצאו ש Unsupervised pre-training גורם באופן ישיר ל Sparseness בשכבות החבויות. בנוסף, ככל שמבצעים Unsupervised pre-training ארוך יותר, כך מתקבל מודל דליל יותר.

חוקרים אלו השוו במאמרם [27] בין רשת עמוקה סטנדרטית שאומנה באמצעות Backprop ורשת עמוקה שמורכבת מ RBMs כלומר שאומנה באמצעות Unsupervised pre-training. הם מדדו והשוו את דלילות המודל בכל סוג רשת. הם מצאו שרשת עמוקה שמורכבת מ RBMs מהווה מודל דליל יותר בהשוואה לרשת ללא Unsupervised pre-training. בנוסף, חוקרים אלו מצאו שככל שה Unsupervised pre-training היה ארוך יותר, כלומר יותר Training epochs, המודל נהיה דליל יותר ויותר.

חשוב לדעת שמודל דליל זה אינדיקציה למודל טוב שמניב מדדי דיוק טובים יותר בשלב המבחן, והתמודדות טובה יותר עם התאמת יתר. נחזור לדון ב Sparsity ו Sparse coding , ומדוע זה קשור להתאמת יתר, בפרק הבא.

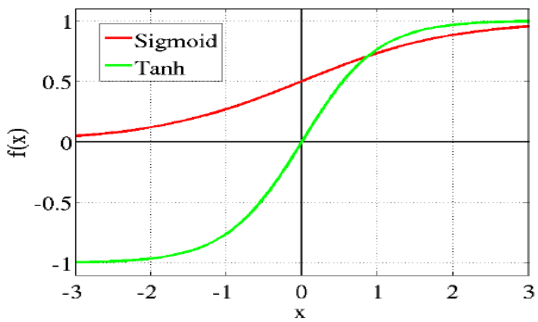
Deep Sparse Rectifier Neural Networks 5.2

[9,15,16,17,30,35,42]

Rectifier 5.2.1

כאן המקום להזכיר מרכיב של רשתות נוירונים שנדון בחלק הראשון של עבודה זו והוא פונקציית אקטיבציה.

פונקציית אקטיבציה מגדירה את הפלט של הנוירון בהינתן הקלט. עד לשנת 2011 פונקציות האקטיבציה השכיחות היו Logistic sigmoid או Tanh כמוצג באיור 23.

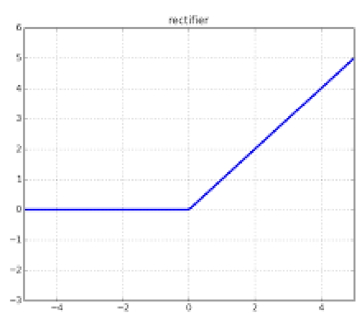


איור 23. פונקציות אקטיבציה שכיחות (Sigmoid & Tanh) עד לשנת 2011.

(*Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics)*

פונקציות אקטיבציה אלו היו שכיחות בגלל המאפיינים המתמטיים שלהם שהתאימו ל Backpropagation algorithm. כפי שפורט והורחב פרק 2.

אך ישנה עוד פונקציית אקטיבציה אפשרית, בשם Rectifier. ההגדרה של הפונקציה מאוד פשוטה:



$f(x)=x \quad x>0$

$f(x)=0 \quad x<=0$

Bengio ושות' מצאו 2 ממצאים עיקריים:

1. רשתות עמוקות עם Rectifier מניבות מדדי דיוק טובים יותר מרשתות עם פונקציות אקטיבציה שכיחות שעברו Unsupervised pre-training כלומר, Rectifier מפחית התאמת יתר טוב יותר מ Unsupervised pre-training
2. Unsupervised pre-training לא משפרת את התאמת יתר ברשתות עם Rectifier, מעבר לשיפור שה Rectifier גורם.

נבצע ניתוח השוואתי בין הביצועים של Rectifier neural networks לרשתות עמוקות עם פונקציות אקטיבציה אחרות. ונבחן את ההשפעה של Unsupervised pre-training על סוגי הרשתות השונות.

5.2.2 השוואה בין Rectifier Neural Network לשיטות אחרות.

מטרות הניסויים [15]: השוואה של רשתות עם פונקציית האקטיבציה Rectifier, לרשתות עם פונקציית האקטיבציה Tanh.

בנוסף, נבחנה ההשפעה של Unsupervised pre-training על כל אחת מהרשתות. כל בסיסי הנתונים שעליהם נעשתה ההשוואה קשורים למשימת זיהוי אובייקטים ו Images. MNIST, בסיס נתונים שכבר הכרנו, לזיהוי ספרות בכתב יד. CIFAR-10 - תמונות צבעוניות של אובייקטים שונים המחולקים ל 10 מחלקות (מטוס, מכונית, צפור, חתול, צבי, כלב, צפרדע, סוס, ספינה, ומשאית). NISTP - ורסיה של NIST עם עיוותים, משימה קשה יותר מNIST. NORB - תמונות של צעצועים מחולקים ל 6 מחלקות.

איור 24 מציג את **תוצאות הניסויים**:

- באופן גורף Rectifier טוב יותר מרשתות עם פונקציית אקטיבציה Tanh
- Unsupervised pre-training מועיל רק לפונקציית אקטיבציה Tanh ולא ל Rectifier
- Rectifier ללא Unsupervised pre-training, טוב יותר מרשתות אחרות עם Unsupervised pre-training

מאמר זה היווה נקודת מפנה במחקר שעוסק ברשתות נוירונים והתאמת יתר. אין צורך יותר ב Unsupervised pre-training, אותו מרכיב שגרם לתחייה מחודשת של רשתות נוירונים עמוקות ב2006, דעך ב2011 , לטובת Rectifier neural networks.

Table 1: **Test error on networks of depth 3.** Bold results represent statistical equivalence between similar experiments, with and without pre-training, under the null hypothesis of the pairwise test with $p = 0.05$.

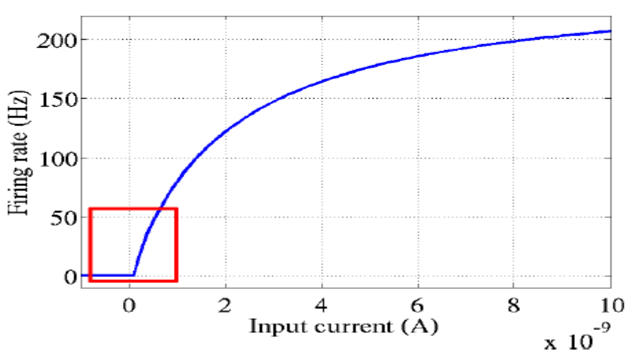
Neuron	MNIST	CIFAR10	NISTP	NORB
With unsupervised pre-training				
Rectifier	**1.20%**	49.96%	**32.86%**	**16.46%**
Tanh	**1.16%**	50.79%	35.89%	17.66%
Without unsupervised pre-training				
Rectifier	**1.43%**	**50.86%**	**32.64%**	**16.40%**
Tanh	1.57%	52.62%	36.46%	19.29%

איור 24. השוואה בין Rectifier ורשתות אחרות והשפעת Unsupervised Pre-Training

(*Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks.*
In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics.)

5.2.3 מדוע ה Rectifier יעילה בהפחתת התאמת היתר ?
[1,2,15,16,17,22,37]

התאמה טובה יותר לפעילות של נירון ביולוגי וsparsity. הייתה מחשבה שהפונקציה הסיגמואידית ודומותיה מתאימות לתיאור פעילות של נוירון ביולוגי בגלל הרציפות וההגעה לרוויה. אך אם נתבונן ברזולוציה גבוהה (Zoom in) נראה שלא כך הדבר. הפונקציה הסיגמואידית מפספסת תכונה מאוד חשובה של הנוירון הביולוגי והוא Sparsity. איור 25 מציג את פעילות הנוירון הביולוגי.



איור 25. פעילות הנוירון הביולוגי.

הפעילות, הפלט, של הנוירון (Firing rate) כפונקציה של הקלט

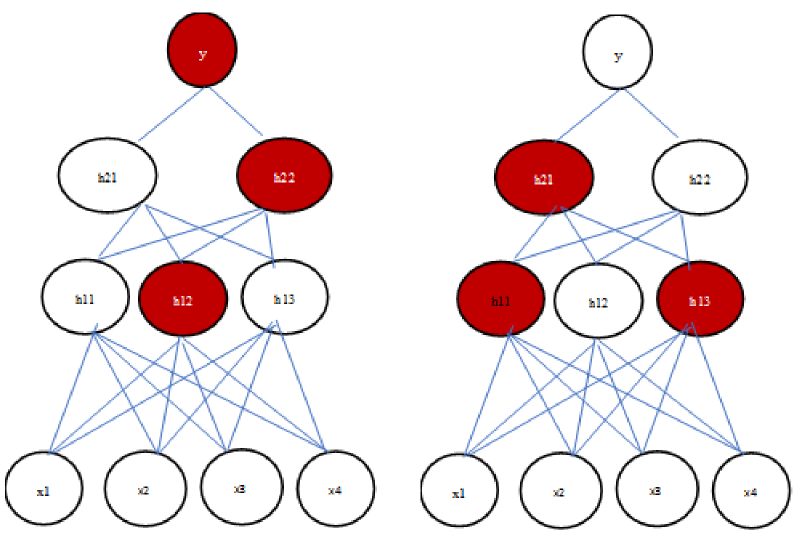
(*Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks*
In Proceedings of the Fourteenth International Conference on Artificial Intelligence and

כשמסתכלים במבט על (Zoom out), אכן פעילות הנוירון מאופיינת ע"י רציפות והגעה לרוויה בהתאמה לפונקציה הסיגמואידית. אולם אם מסתכלים ברזולוציה גבוהה יותר (Zoom in) ריבוע אדום, באיור 34. רואים שהנוירון מאופיין ע"י פעילות Sparsity. מתבטא בכך שכאשר x<=0 הנוירון שותק ולא מגיב. ועבור 0<x<1.5 הנוירון

מאופיין בפעילות שהיא בקרוב ליניארית. ניתן להשוות גרף זה, של פעילות נוירון ביולוגי, מאיור 25, ולראות את ההתאמה עם ה rectifier המתואר בעמודים הקודמים.

המוח הביולוגי מבצע Sparse coding. בהינתן אוכלוסייה של נוירונים המטפלת בגירויים מסוג מסוים. רק תת קבוצה ספציפית של נוירונים מסך כל הנוירונים תגיב עבור כל גירוי ספציפי. זו הדרך של המוח לקודד אינפורמציה וזו הסיבה שהמוח הביולוגי יעיל מאוד בביצוע קלסיפיקציות. לצורך הפשטה, נציג את הדוגמא הבאה, כמובן שבמוח הדבר מורכב הרבה יותר וזה רק לצורך הדגמת העיקרון. נגיד ויש 10 נוירונים שמגיבים לריח (נוירון1, נוירון2, נוירון3 וכו'). ונגיד שיש 50 סוגי ריחות (ריח1, ריח2, ריח3 וכו'). ***עבור ריח1***: נוירון5 ונוירון9 פעילים, שאר הנוירונים שותקים. ***עבור ריח2***: נוירון4 ונוירון6 פעילים השאר שותקים. ***עבור ריח3***: נוירון8, ונוירון10 פעילים השאר שותקים ועל זו הדרך בשאר הריחות. זו דרך מאוד יעילה לבצע קלסיפיקציה. חוסר חפיפה בין ריחות שונים בפעילות של הנוירונים מאפשר למוח לזהות בדיוק רב ובקלות גם ריחות שמאוד דומים זה לזה. רק באמצעות הידיעה איזו תת קבוצה פעילה ניתן לדעת במדויק את הריח המתאים. וזו בדיוק המטרה שלנו ברשתות נוירונים עמוקות! לבנות מודל שיאפשר ביצוע של קלסיפיקציה מדויקת.

איור 26 מציג באופן פשטני יתרון של Sparsity.



יתרון של Sparsity. לרשת מוצגת 2 דוגמאות קלט שונות (איור שמאלי ואיור ימני). עבור כל קלט רק תת קבוצה ספציפית מתוך סך כל הנוירונים פעילה כתגובה לקלט ספציפי. ניתן לבצע קלסיפיקציה מדויקת רק מלדעת איזו תת קבוצה ספציפית פעילה.

בשנת 2012, הינטון ושות' גילו טכניקה חדשה שמצליחה להפחית עוד יותר את בעיית התאמת יתר ולשפר את יכולת ההכללה של הרשת. לטכניקה זו קוראים Dropout.

Dropout 5.3
[3,9,18,23,36,38,41,43]

5.3.1 תיאור הטכניקה
בשנת 2012 הינטון ושות' [18,38] הציעו טכניקה חדשה ומעניינית ששמה Dropout.

שלב האימון: בכל שלב במהלך האימון מוחקים באופן זמני כל נוירון מהשכבות הנסתרות בהסתברות מסוימת, ברוב המקרים והרשתות הסתברות של 0.5 קרובה לאופטימלית. ערך זה נמצא אופטימלי באמצעות ניסוי וטעייה של ערכים שונים. כמו כן ניתן למחוק חלק מיחידות הקלט. מריצים Backprop ומעדכנים את הפרמטרים (משקולות) ברשת. לאחר מכן, מחזירים את הנוירונים שהוסרו וחוזרים שוב על התהליך, כלומר מוחקים באופן זמני כל נוירון בהסתברות 0.5 (או הסתברות אחרת) מריצים Backprop ומעדכנים את הפרמטרים ברשת וכן הלאה.

איור 27 משווה בין ארכיטקטורה סטנדרטית אל מול השפעת השיטה Dropout על ארכיטקטורת הרשת. אם ברשת יש H נוירונים בשכבות החבויות, הרי שבמהלך ה Dropout עם הסתברות השמטה רנדומלי של 0.5 לכל נוירון בשכבה החבויה, אנו למעשה דוגמים כל פעם ארכיטקטורה רנדומלית מ 2^H ארכיטקטורות אפשריות. ישנן סך הכול 2^H אפשרויות לארכיטקטורות שונות.

שלב המבחן: משתמשים **בכל הנוירונים ובכל המשקולות של הרשת** אחרי שמחלקים ב 2 את ערכי כל המשקולות. באופן כללי, בזמן המבחן, משתמשים בכל הנוירונים והמשקולות אחרי שמכפילים את ערכי כל המשקולות בהסתברות, שהנוירון נשאר ברשת. כמתואר באיור 28. כאשר ההסתברות להישארות היא 0.5 מכפילים ב 0.5 (מחלקים ב 2). הליך זה גורם לכך שהרשת בזמן המבחן מחשבת בדיוק את הממוצע הגיאומטרי של החיזויים מכל ה 2^H מודלים.

Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov

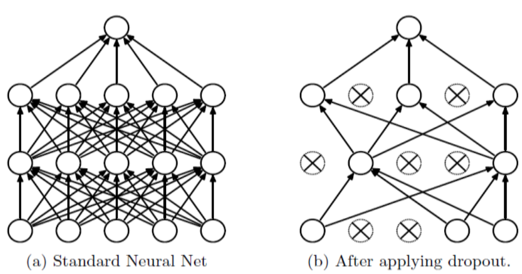


Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

איור 27. השפעה של Dropout על הרשת. מוחקים חלק רנדומלי מהנוירונים מבצעים Backpropagation מעדכנים פרמטרים, מחזירים את הנוירונים וחוזרים שוב על ההליך.

(Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.)

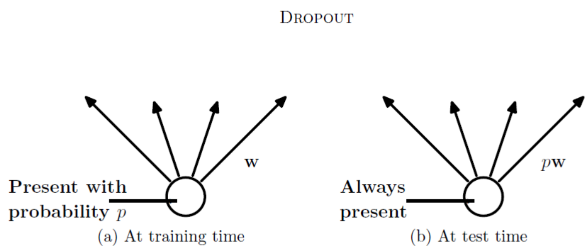


Figure 2: **Left**: A unit at training time that is present with probability p and is connected to units in the next layer with weights **w**. **Right**: At test time, the unit is always present and the weights are multiplied by p. The output at test time is same as the expected output at training time.

איור 28. Dropout. במהלך האימון מוחקים באופן זמני חלק מהנוירונים מבצעים Backpropagation מעדכנים פרמטרים, מחזירים את הנוירונים וחוזרים שוב על ההליך. בשלב המבחן משתמשים בכל הנוירונים אחרי שמכפילים כל משקולת בp (הסתברות להישאר ברשת).

(Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.)

Dropout, מסייע בהפחתת התאמת היתר, על ידי מניעת Co-adaptation ("התרגלות") בין נוירונים בזמן האימון, עניין שיוסבר בסעיף המסביר את הסיבות ליעילות של Dropout. בנוסף, ניתן להתייחס ל Dropout כסוג של Model averaging, כלומר בניית הרבה מודלים (רשתות), אימון הרשתות באופן נפרד על דוגמאות האימון, ולקיחת הממוצע של הרשתות בשלב המבחן. מכיוון ש Dropout גורם לכך שהרשת בזמן המבחן מחשבת בדיוק את הממוצע הגיאומטרי של החיזויים מכל ה 2^H מודלים אפשריים. שיטה זו יעילה בהפחתת התאמת יתר ומשפרת את ביצועי הרשת. התיאוריה מאחורי היעילות של Dropout תדון ביותר פירוט, בהמשך, בפרק המיועד לכך. כעת נעבור להשוואה בין ביצועי רשתות עם ובלי Dropout.

5.3.2 Dropout vs Without Dropout

[15,

18,38]

מטרה: השוואה בין ביצועי רשתות עם Dropout לרשתות ללא Dropout.

הניסוי: השוואת ביצועי רשתות עם Dropout אל מול רשתות ללא Dropout על בסיס נתונים MNIST. כפי שכבר הוזכר MNIST הוא בסיס נתונים מאוד שכיח בעבודות של למידת מכונה. זהו בסיס נתונים של ספרות שנכתבו בכתב יד. בסיס הנתונים מכיל 60,000 דוגמאות אימון ו 10,000 דוגמאות מבחן. היתרון הגדול שבהצגת תוצאות שבוצעו על MNIST הוא שאנו יכולים לא רק להציג תוצאות של המחקר הנוכחי של הינטון ושות' שבו אנו דנים אלא גם להשוות את התוצאות לאנליזות קודמות שביצענו בפרקים הקודמים.

תוצאות:

Dropout משפר את ביצועי הרשת ומפחית התאמת יתר באופן מאוד משמעותי. עפ"י המחקר של הינטון ושות', ללא Dropout הרשתות מניבות מעל ל 1.5% טעות על MNIST כאשר רשתות עם Dropout מתקרבות ל 1 % טעות. איור 29 מציג השוואה גרפית על בסיס נתונים, MNIST, Dropout vs Without Dropout. ציר y מייצג את אחוז הטעות, ציר x מייצג את התקדמות הליך עדכון המשקולות ב Backprop. הגרפים העליונים מייצגים רשתות ללא Dropout והגרפים התחתונים מייצגים רשתות עם Dropout. על כן הינטון ושות' מסיקים ש Dropout יעיל בהפחתת בעיית התאמת היתר.

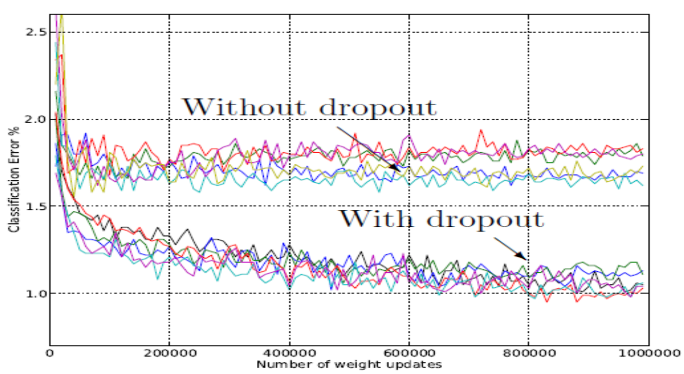


Figure 4: Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

איור 29. השוואה ויזואלית בין רשתות עם Dropout לרשתות בלי Dropout על MNIST

(Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.)

איור 30 מציג השוואה מפורטת בין סוגי רשתות עם טכניקות שונות והביצועים שלהן על MNIST.

6.1.1 MNIST

Method	Unit Type	Architecture	Error %
Standard Neural Net (Simard et al., 2003)	Logistic	2 layers, 800 units	1.60
SVM Gaussian kernel	NA	NA	1.40
Dropout NN	Logistic	3 layers, 1024 units	1.35
Dropout NN	ReLU	3 layers, 1024 units	1.25
Dropout NN + max-norm constraint	ReLU	3 layers, 1024 units	1.06
Dropout NN + max-norm constraint	ReLU	3 layers, 2048 units	1.04
Dropout NN + max-norm constraint	ReLU	2 layers, 4096 units	1.01
Dropout NN + max-norm constraint	ReLU	2 layers, 8192 units	0.95
Dropout NN + max-norm constraint (Goodfellow et al., 2013)	Maxout	2 layers, (5 × 240) units	0.94
DBN + finetuning (Hinton and Salakhutdinov, 2006)	Logistic	500-500-2000	1.18
DBM + finetuning (Salakhutdinov and Hinton, 2009)	Logistic	500-500-2000	0.96
DBN + dropout finetuning	Logistic	500-500-2000	0.92
DBM + dropout finetuning	Logistic	500-500-2000	**0.79**

Table 2: Comparison of different models on MNIST.

איור 30. השוואה בין ביצועי רשתות שונות עם Dropout לרשתות בלי Dropout על MNIST

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.

עפ"י הינטון ושות' רשת סטנדרטית ללא Dropout ועם פונקציית אקטיבציה סיגמואידית (Logistic) מניבה 1.6% טעות. רשת עם Dropout ופונקציית אקטיבציה סיגמואידית מניבה 1.35% טעות. רשת עם Dropout ו Rectifier מניבה 0.95% טעות. בעבודה הזו של הינטון ושות' חסרה היעילות של רשת עם Rectifier ללא Dropout. אך ניתן לחזור למחקר של Glorot, Bordes, & Bengio (2011) שם מובא ש Rectifier ללא Dropout מניבה 1.43% טעות. רשת שעברה Unsupervised pre-training ביחד עם Dropout מניבה 0.79% טעות. רשת שעברה Unsupervised pre-training ללא Dropout מניבה 0.96% טעות. **לסיכום, בברור נראה ש Dropout משפרת את ביצועי הרשת ומפחיתה משמעותית את התאמת היתר. Dropout משפרת את ביצועי הרשת גם בשילוב עם Rectifier וגם בשילוב עם Unsupervised pre-training.** כאשר על פי מאמר זה Dropout בשילוב עם Unsupervised pre-training עדיף על השילוב של Dropout עם Rectifier.

בנוסף, הינטון ושות' מביאים במאמרם השוואה בין רשתות עם Dropout ו Without dropout על בסיסי נתונים שונים:

- Street view house numbers(SVHN) - אלו תמונות מ Google street והמשימה היא לזהות את מספר הבתים.
- CIFAR-10 ו CIFAR100 - זיהוי אובייקטים המחולקים ל 10 ו 100 קטגוריות בהתאמה.

- ImageNet - בסיס נתונים ענק המכיל מעל 15 מיליון תמונות המחולק ל 22,000 קטגוריות. נחזור לבסיס נתונים זה בהמשך.
- TIMIT - בסיס נתונים לזיהוי קולי (Speak recognition)

באופן גורף, Dropout שיפר את ביצועי הרשת בפתרון המשימות בכל בסיסי הנתונים לעיל, והפחית את בעיית התאמת יתר.

5.3.3 מדוע Dropout מפחית התאמת יתר ברשתות עמוקות?

[18,23,32,38]

ישנן מספר הסברים מדוע Dropout מפחית את בעיית התאמת יתר:

1. הפרספקטיבה הראשונה היא שבאופן הפשוט ביותר Dropout הוא גם Model averaging. Model averaging היא שיטה ידועה עשרות שנים לפני Dropout, שבה מבצעים Mixture of experts. אם רשת גדולה מובילה להתאמת יתר, ניתן להקטין את הרשת. אבל לבנות מספר רשתות שונות (ארכיטקטורה שונה, פונקציות אקטיבציה שונות, ערכי משקולות התחלתיים שונים וכו') כך שכל רשת תייצר מודל באופן עצמאי על דוגמאות האימון. כלומר ב Model averaging בונים ומאמנים הרבה רשתות שונות על דוגמאות האימון.

ב Model averaging, בזמן המבחן נותנים כפלט את הממוצע המתקבל מכל הרשתות עבור דוגמת המבחן הספציפית. זו שיטה ותיקה שיכולה להפחית התאמת יתר.

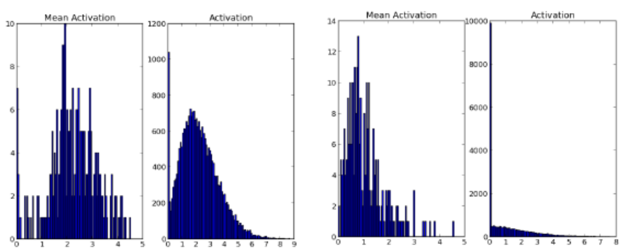
במובן מסוים Dropout הוא גם Model averaging. מכיוון שב Dropout כל פעם אנו דוגמים רשת שונה מתוך 2^H רשתות (ארכיטקטורות) אפשריות (H מספר הנוירונים ברשת). ומאמנים כל רשת שדגמנו ומשנים את הפרמטרים בהתאם. בסוף, בשלב המבחן אנו מניבים חיזוי עם הרשת השלמה. חיזוי זה הוא בהתאם לממוצע שהיה מתקבל מכל ה 2^H רשתות. כאשר משתמשים בכל הנוירונים והמשקולות, מוכפלים ב p (ההסתברות של יחידה לא להישאר בכל שלב) מקבלים את הממוצע הגיאומטרי של הפלט שהיה מתקבל מכל הרשתות וזהו סוג של Model averaging.

2. מניעת Co-adaptation בין נוירונים. במהלך ה Backpropagation algorithm הנגזרת המתקבלת מכל פרמטר אומרת לפרמטר כיצד עליו לשנות את ערכו כך שערך פונקציית המחיר תופחת, ***בהינתן*** הערך של כל שאר הפרמטרים. כלומר, פרמטר יכול להשתנות בהסתמך על טעות שמבצעים פרמטרים אחרים. הפרמטר לומד איך להשתנות כך שפונקציית המחיר תפחת בהינתן ערכי המשקולות האחרים שייתכן והם ערכים שגויים. זה גורם לקיבוע הטעות של פרמטרים אחרים ומוביל ל Co-adaptation בין נוירונים. Co-adaptation מובילה להתאמת יתר כי ה Co-adaptation עובד רק בדוגמאות האימון והיא ללא יכולת הכללה לדוגמאות חדשות. ב Dropout כל יחידה חייבת להיות עצמאית, ולא יכולה להסתמך על ערכי משקולות אחרים, כי כל פעם היחידה צריכה לתפקד בארכיטקטורה שונה עם משקולות שונות.

3. יצירת Sparsity. מעניין לגלות ש ל Dropout יש השפעה על Sparsity למרות שהשיטה כלל לא מכוונת לכך. Dropout גורם לפעילות דלילה של הנוירונים בשכבות החבויות. כבר הזכרנו ש ל Sparse coding יש תפקיד מפתח ביכולת של המוח הביולוגי, לבצע קלסיפיקציות מדויקות. מסתבר שבלי כוונה ליצור Sparsity, Dropout גורם למודל דליל.

איור 31 מציג את תוצאות התצפית של הינטון ושות' לגבי ההשפעה של Dropout על Sparsity. בסיס הנתונים שעליו הרשת התאמנה הוא MNIST. הינטון ושות' בחנו את הפעילות של יחידות הנוירונים לאחר סיום אימון הרשתות. והשוו בין הפעילות של יחידות הנוירונים ברשתות שאומנו עם Dropout אל מול רשתות שאומנו ללא Dropout.

7.2 Effect on Sparsity



(a) Without dropout (b) Dropout with $p = 0.5$.

Figure 8: Effect of dropout on sparsity. ReLUs were used for both models. **Left**: The histogram of mean activations shows that most units have a mean activation of about 2.0. The histogram of activations shows a huge mode away from zero. Clearly, a large fraction of units have high activation. **Right**: The histogram of mean activations shows that most units have a smaller mean mean activation of about 0.7. The histogram of activations shows a sharp peak at zero. Very few units have high activation.

איור 31. השפעה של Dropout על Sparsity

(Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.)

בעוד שהממוצע הכללי של פעילות הנוירונים הוא 2.0 ללא Dropout הממוצע הכללי הוא 0.7 כשמוסיפים Dropout. בנוסף פיזור הפעילות שונה לחלוטין. עם Dropout רוב הנוירונים היו מושתקים, פעילות 0, עבור רוב דוגמאות האימון. בעוד שללא Dropout לרוב הנוירונים פעילות גבוהה עבור רוב דוגמאות האימון.

ImageNet Large Scale Visual Recognition Challenge.6

[16,17,25,26,38]

ImageNet 6.1

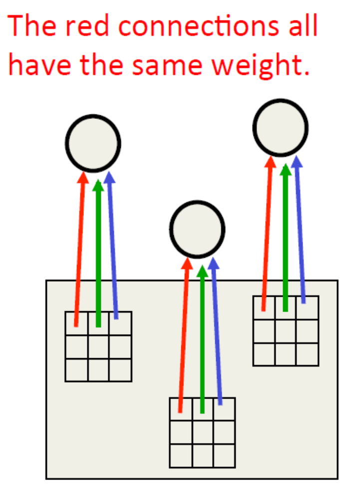
[17,24,38]

בסיס נתונים זה מכיל 15 מיליון תמונות ברזולוציה גבוהה, כאשר כולן מתויגות (Labeled). התמונות שייכות ל 20,000 קטגוריות. משנת 2010 כל שנה, התבצעה תחרות בשם ImageNet Large Scale Visual Recognition Challenge, שבו קבוצות המחקר מתחרות עם הרשתות שלהן על בסיס נתונים זה. מכיוון שיש 20,000 קטגוריות אחוז הטעות מחולק ל 2. Top-1: אחוז הדוגמאות שבהן הרשת לא בחרה בקלסיפיקציה הנכונה, כבחירה הראשונה של הרשת. Top-5: אחוז הדוגמאות שהרשת לא בחרה בקלסיפיקציה הנכונה, באף אחת מתוך חמשת הקלסיפיקציות עם ההסתברות הגבוהה ביותר שהרשת נתנה.

יש 4 רשתות מרכזיות לזיהוי אובייקטים, שפותחו משנת 2012 לאחר גילוי הטכניקות להפחתת התאמת יתר שנדונו בעבודה זו.

1. AlexNet, הינטון ושות', בשנת 2012
2. Visual Geometry Group (VGG) Net, קבוצת מחקר מאוקספורד, בשנת 2014
3. GoogleNet, חברת גוגל, בשנת 2014
4. ResNet, מיקרוסופט, בשנת 2015

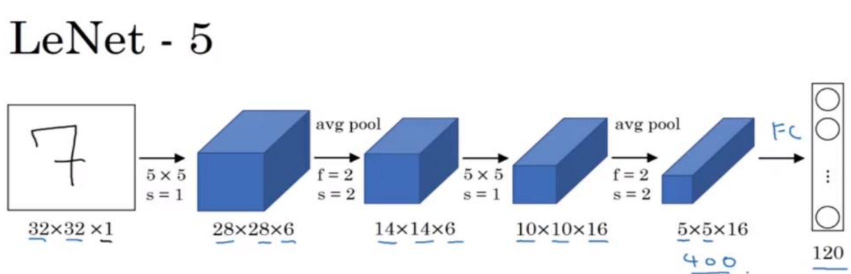
יש לציין שכל הרשתות הללו הן מסוג ConvNet היעילות שלהן נובעות מארכיטקטורה מתאימה לזיהוי אובייקטים תוך כדי הפחתה משמעותית במספר הפרמטרים החופשיים המתבצעת באמצעות Weight sharing. משתמשים באותן משקולות עבור חלקים שונים של התמונה, מוצג באיור 32 ואיור 33.



איור 32. Weight Sharing

Neural Networks for Machine Learning by University of Toronto (Geoffrey Hinton)

הרשת העמוקה הראשונה שהצליחה להניב ביצועים פרקטיים בזיהוי אובייקטים היא LeNet מסוג ConvNet שפותחה בשנת 1998. ללא Unsupervised pre-training, Rectifier או Dropout.



.

איור 33. ארכיטקטורה של LeNet

(Deeplearning.ai)

לא ניתן במסגרת עבודה זו להיכנס לפרטים הטכניים של ConvNets. אך בכל זאת ארצה להביא את הארכיטקטורה של הרשת באופן כללי. איור 33 מציג את הארכיטקטורה הכללית של LeNet. באופן כללי, הסימון מתחת לחצים: 5x5 מסמל פילטר בגודל של 5x5 ש "צועד או מוזז " על גבי הפיקסלים. גודל הצעד או התזוזה הוא בהתאם לגודל שקובעים כStride (מסומן באות s). גודל 5x5 משמעותו רק 25 משקולות שונות שמוזזות על גבי התמונה (פילטר). לחלקים שונים של התמונה אותם המשקולות (Weight sharing). ההיגיון הוא שאם לדוגמא אני רוצה לזהות קצוות בתמונה (edges) אז אותן המשקולות שמזהות קצוות בחלק אחד של התמונה יהיו טובות לזיהוי קצוות בחלק אחר של התמונה.

עם התפתחות הטכניקות להפחתת התאמת יתר, שתוארו ונותחו בעבודה זו, קבוצות מחקר החלו לפתח רשתות יעילות יותר ויותר בזיהוי אובייקטים המתבססות על הטכניקות שתוארו בספר זה. בסיס הנתונים שעליו קבוצות המחקר השונות עבדו ובדקו את ביצועי הרשת, היה זהה. הקבוצות כולן עבדו ובדקו את הרשתות שלהם על ImagNet.

AlexNet 6.2

[38]

רשת זו דומה לLeNet שפותחה ב1998. AlexNet הניבה תוצאות מרשימות מאוד על ImageNet כמפורט באיור 34.

SRIVASTAVA, HINTON, KRIZHEVSKY, SUTSKEVER AND SALAKHUTDINOV

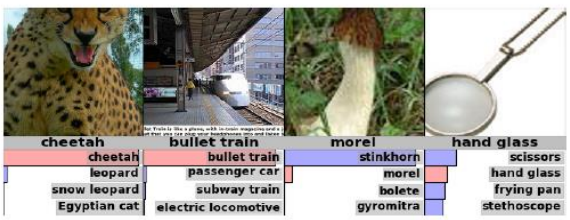


Figure 6: Some ImageNet test cases with the 4 most probable labels as predicted by our model. The length of the horizontal bars is proportional to the probability assigned to the labels by the model. Pink indicates ground truth.

Model	Top-1	Top-5
Sparse Coding (Lin et al., 2010)	47.1	28.2
SIFT + Fisher Vectors (Sanchez and Perronnin, 2011)	45.7	25.7
Conv Net + dropout (Krizhevsky et al., 2012)	37.5	17.0

Table 5: Results on the ILSVRC-2010 test set.

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SVM on Fisher Vectors of Dense SIFT and Color Statistics	-	-	27.3
Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT	-	-	26.2
Conv Net + dropout (Krizhevsky et al., 2012)	40.7	18.2	-
Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)	38.1	16.4	16.4

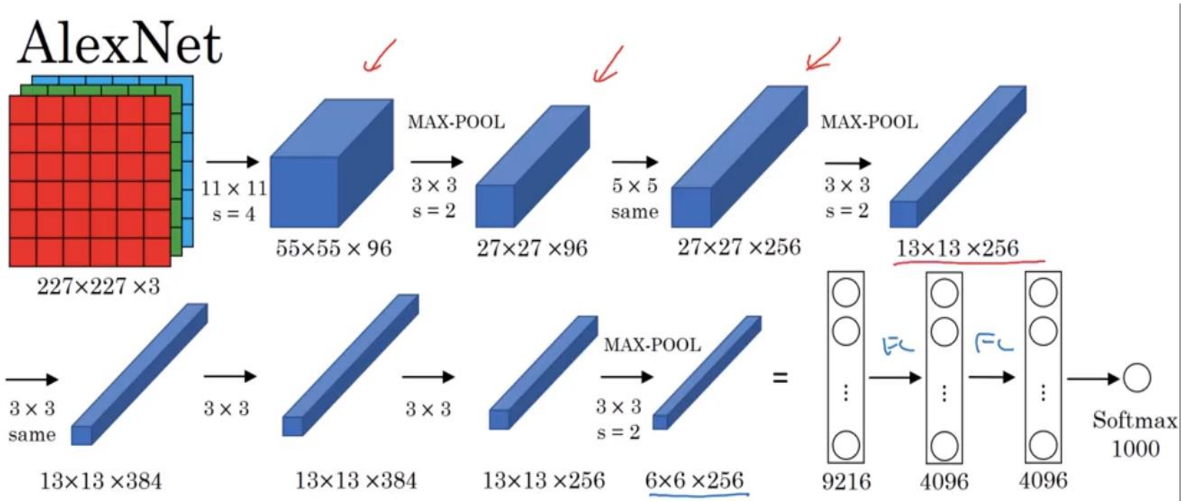
Table 6: Results on the ILSVRC-2012 validation/test set.

איור 34. AlexNet on ImageNet

(Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929-1958.)

סך הכול ב AlexNet יש 8 שכבות חבויות והגיעו לכ 16.4% טעות. שיפור משמעותי משנה קודמת שבה הרשת הטובה ביותר הגיעה ל 26% טעות.

איור 35 מציג את הארכיטקטורה של AlexNet. מבחינה מבנית AlexNet דומה ל LeNet. החידוש הוא שב AlexNet **יש יותר שכבות חבויות**. השתמשו בDropout, ובפונקציית האקטיבציה Rectifier. טכניקות שנדונו בעבודה זו. בנוסף AlexNet אומנה על מעבד GPU שהינו יעיל יותר מ CPU להרצת Backprop.



איור 35. ארכיטקטורה של AlexNet

(Deeplearning.ai)

VGG Net & GoogleNet 6.3
[17]

בשנת 2014 חוקרים הצליחו לשפר את הביצועים של AlexNet. קבוצת חוקרים מאוקספורד פיתחה את הרשת VGG עם 19 שכבות חבויות. הרשת הגיעה ל 7.32% טעות על ImageNet. באותה שנה Google פיתחה את GoogleNet עם 22 שכבות חבויות שהגיעה ל 6.67% טעות על ImageNet. כל הרשתות הן כמובן ConvNet.

רשת VGG מאופיינת ע"י כך שיש חזרה

ResNet 6.4
[17]

בשנת 2015 הגיעה פריצת דרך נוספת משמעותית ב ImageNet Large Scale Visual Recognition Challenge חוקרים ממיקרוסופט הצליחו ליצור רשת מאוד עמוקה, עם 152 שכבות חבויות, תוך שמירה על מספר פרמטרים כולל נמוך. כך ניתן ליהנות מהיתרונות של עיבוד על פני שכבות רבות של יחידות לא לינאריות. כלומר, ליצור מודל מאוד אקספרסיבי, תוך שמירה על מספר פרמטרים כולל נמוך ומניעת התאמת יתר. הטריק שלהם היה שהקישוריות בין הנוירונים לא יהיה רק בין שכבות סמוכות כמו הארכיטקטורות הסטנדרטיות אלא שחלק מהנוירונים במקום להתחבר לשכבה הסמוכה מדלגים על שכבות ומתחברים לנוירונים בשכבות עמוקות יותר. הם עיצבו ארכיטקטורה מאוד עמוקה ושמרו על מורכבות יחסית נמוכה. **הרשת שלהם הגיעה ל 4.94% טעות**, תוך שימוש בטכניקות לפתרון בעיית

התאמת יתר שנדונו בעבודה זו (Rectifier & Dropout). **ממוצע הטעות (top-5) של אנשים על ImageNet הוא 5.1%.**

איור 36 מציג את תוצאות ResNet מול VGG ו GoogleNet. רשת VGG מניבה 7.32 אחוז טעות, GoogLeNet מניבה 6.66 אחוז טעות. ResNet מניבה 4.94 אחוז טעות על אותו בסיס נתונים.

.

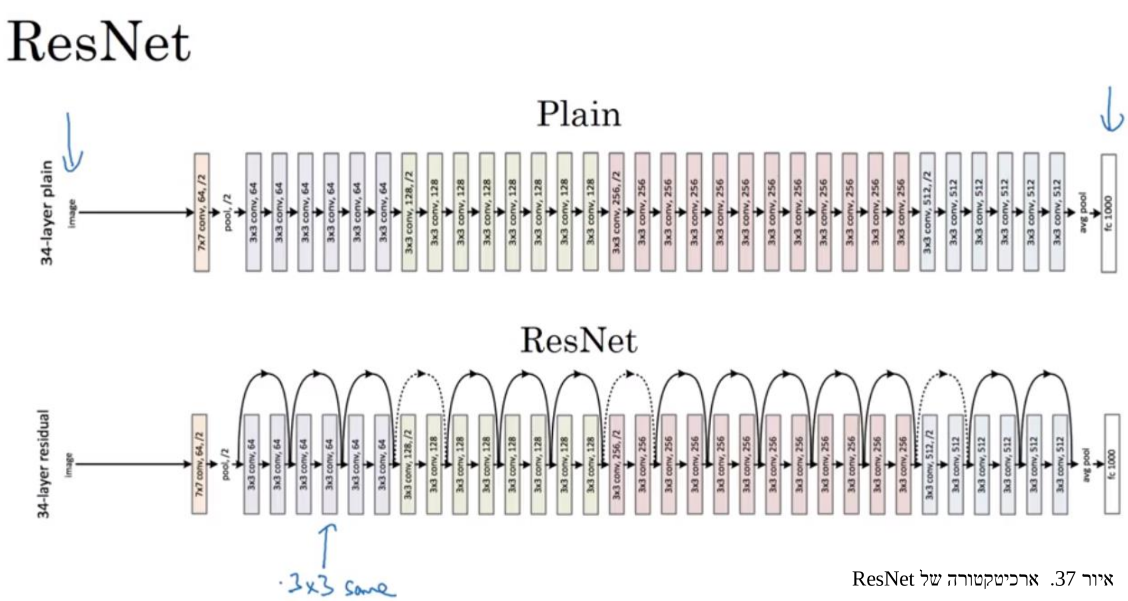
	method	top-5 (**test**)
in ILSVRC 14	SPP [12]	8.06
	VGG [29]	7.32
	GoogLeNet [33]	6.66
post ILSVRC 14	VGG [29] (arXiv v5)	6.8
	ours	**4.94**

Table 7. **Multi-model** results for the ImageNet 2012 test set.

איור 36. תוצאות ResNet מול VGG ו GoogleNet

(He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*)

איור 37 מציג את הארכיטקטורה הייחודית של ResNet. ניתן לראות שמשקולות משכבות קודמות מדלגות על שכבות ומתחברות לשכבות עמוקות יותר. בנוסף, הם שמרו על מורכבות רשת קטנה והשתמשו בעקביות בפילטר 3x3.



איור 37. ארכיטקטורה של ResNet

(Deeplearninig.ai)

איור 38 מציג דוגמא לקלסיפיקציות של ResNet על ImageNet



Figure 5. Example validation images successfully classified by our method. For each image, the ground-truth label and the top-5 labels predicted by our method are listed.

איור 38. דוגמאות לקלסיפיקציות של ResNet על ImageNet

(He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*)

מעניין לראות את המגמה המשתקפת מהסקירה שבוצעה בפרק זה של העבודה. להוספת עוד ועוד שכבות חבויות, יש חשיבות קריטית בשיפור הביצועים של רשתות עמוקות בבעיות מורכבות של אינטליגנציה מלאכותית, כדוגמת זיהוי תמונות. מהסקירה לעיל נובע שהוספת יותר ויותר שכבות חבויות קריטית לביצועים הולכים ומשתפרים של רשתות עמוקות לאורך השנים. AlexNet שפותחה בשנת 2012 עם 8 שכבות חבויות, הניבה 16.4% טעות. לאחר מכן ב2014, VGG עם 19 שכבות חבויות, הניבה 7.32% טעות. GoogleNet מ 2014, עם 22 שכבות חבויות, הניבה 6.67 % טעות, ו ResNet מ 2015 עם 152 שכבות חבויות, הניבה 4.94 % טעות, ומתעלה על ביצוע אנושי ממוצע. ***לעומק הרשת יש חשיבות קריטית בשיפור הביצועים***.

בפרק הבא, אציע שיטה חדשה לפתרון בעיית התאמת יתר, המבוססת על מסקנות עבודה זו שמפורטות בתחילת הפרק הבא.

7. הצעה לטכניקה חדשה לפתרון בעיית התאמת יתר ולעיצוב רשתות עמוקות מאוד

מהסקירה שבוצעה בעבודה מסכמת זו עולות מספר **מסקנות**:

1. ככל שהרשת עמוקה יותר כך מחריפה בעיית התאמת יתר

2. שלושת הטכניקות העיקריות שמפחיתות התאמת יתר שסוקרו בעבודה זו
(Unsupervised pre-training, Deep sparse rectifier neural networks, & Dropout)
גורמות למודל דליל (Sparsity) למרות ש Unsupervised pre-training ו Dropout אינן מכוונות כלל ליצירת מודל דליל.

3. לעומק הרשת יש חשיבות קריטית בשיפור הביצועים כפי שמשתקף מהשיפור המתמשך בביצועים לאורך השנים מ AlexNet בעלת 8 שכבות חבויות ועד ל ResNet עם 152 שכבות חבויות.

על כן יש צורך בטכניקה שמחד תאפשר עיצוב של רשתות עמוקות מאוד ומאידך תפחית את בעיית התאמת יתר שמתעוררת עם עלייה במספר השכבות והפרמטרים.

7.1 מבוא קצר לטכניקה

[2, 8, 14, 16, 19, 27, 40]

הטכניקה המוצעת כאן מיישמת עקרונות נוירו-ביוכימיים חישוביים על מנת לאפשר יצירת מודל מאוד עמוק ודליל. אביא בקצרה עקרונות אלו בכדי להבין את הטכניקה.

קידוד דליל והתאמת יתר. כפי שהסקנו בעבודה המסכמת, שלושת הטכניקות:
Unsupervised pre-training, Deep sparse rectifier neural networks, & Dropout, גורמות לפעילות דלילה של הנוירונים בשכבות החבויות. קידוד דליל (Sparse coding) הוא עקרון נוירו-ביוכימי המאפיין את פעילות הנוירונים במוח והמאפשר למוח לבצע אבחנות וסיווגים מדויקים מאוד .

גיזום סינפטי וקידוד דליל במוח הביולוגי. במוח הביולוגי הליך הקרוי גיזום סינפטי (Synaptic pruning) מעודד קידוד דליל וחיוני ללמידה.

למידה לא מונחית וגיזום סינפטי במוח. מנגנון של למידה לא מונחית הביאנית (Hebbian unsupervised learning) קובע את החוזק הסינפטי ומקדים את הגיזום הסינפטי. סינפסות חזקות (בעקבות המנגנון ההביאני), נשארות וסינפסות חלשות נגזמות. למידה הביאנית היא הבסיס ללמידה לא מונחית (Unsupervised learning) ברשתות נוירונים מלאכותיות.

גיזום משקולות. גיזום משקולות (Weight pruning) , טכניקה המיושמת ברשתות נוירונים מלאכותיות בהשראת הגיזום הסינפטי. בטכניקה זו מוסרות חלק מהמשקולות במהלך ה Backprop. טכניקה זו יעילה במידה מסוימת אך לא מספיק טובה לפתרון התאמת יתר. הטכניקה הזו עד היום מתבצעת לאחר תחילת ה Backprop.

7.2 תיאור הטכניקה

הטכניקה המוצעת כאן מיישמת גיזום של משקולות (Weight pruning) באופן חדשני, **<u>לפני</u>** התחלת ה Backprop לשם עיצוב רשתות עמוקות מאוד ודלילות. שני מרכיבים חיוניים ביותר (עפ"י עבודה מסכמת זו) לבניית מודלים עם ביצועים משופרים במשימות עם מורכבות שהולכת וגדלה.

תיאור הטכניקה:

1. התחל עם רשת עמוקה מאוד (מאות ואלפי שכבות חבויות) עם קישוריות מלאה.

2. בצע למידה בלתי מונחית מהירה.

3. הסר את המשקולות עם הערך המוחלט הנמוך ביותר.

4. בצע Backprop עם הרשת הגזומה.

טכניקה זו נועדה להשפיע ישירות על דלילות המודל. בנוסף, טכניקה זו נועדה לאפשר עיצוב רשתות עמוקות מאוד בגלל הליך הגיזום והסרת המשקולות שמתבצע לפני התחלת ה Backprop.

הטכניקה עקבית עם הסדר שמתבצע במוח הביולוגי: למידה בלתי מונחית ולאחריה גיזום סינפטי. כמו כן הטכניקה עקבית עם הסדר המבני-דינמי שמתבצטא במוח במהלך החיים. המוח מתחיל עם חיבוריות גבוהה מאוד ומאוחר יותר מתבצעת הסרה משמעותית של סינפסות בגיל ההתבגרות. זאת בכדי לשפר פונקציונליות של המוח.

8. רשימת מקורות

[1] Arora, S., Ge, R., Ma, T., & Moitra, A. (2015). Simple, efficient, and neural algorithms for sparse coding. *arXiv preprint arXiv:1503.00778*.

[2] A sparse memory is a precise memory. Oxford Science blog. 28 Feb 2014. http://www.ox.ac.uk/news/science-blog/sparse-memory-precise-memory

[3] Ba, J., & Frey, B. (2013). Adaptive dropout for training deep neural networks. In *Advances in Neural Information Processing Systems* (pp. 3084-3092).

[4] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137-1155.

[5] Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, *2*(1), 1-127

[6] Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

[7] Computational Neuroscience by University of Washington

[8]Crochet, S; Poulet, JFA; Kremer, Y; Petersen, CCH (2011). "Synaptic mechanisms underlying sparse coding of active touch". *Neuron*. 69: 1160–1175.

[9] Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8609-8613). IEEE.

[10] Dayan, P., & Abbott, L. F. (2001). Theoretical neuroscience (Vol. 806). Cambridge, MA: MIT Press.

[11] De Castro, L. N. (2006). Fundamentals of natural computing: basic concepts, algorithms, and applications. *CRC Press.*

[12] Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, *7*(3–4), 197-387.

[13] Erhan, D., Manzagol, P. A., Bengio, Y., Bengio, S., & Vincent, P. (2009). The difficulty of training deep architectures and the effect of unsupervised pre-training. In *International Conference on artificial intelligence and statistics* (pp. 153-160).

[14] Geoffrey E. Hinton (2007) Boltzmann machine. Scholarpedia, 2(5):1668.

[15] Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (pp. 315-323).

[16] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

[17] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026-1034).

[18] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[19] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

[20] Hinton, G. E. (2007). To recognize shapes, first learn to generate images. *Progress in brain research*, *165*, 535-547.

[21]Hinton, G. E., Dayan, P., Frey, B. J., & Neal, R. M. (1995). The" wake-sleep" algorithm for unsupervised neural networks. *Science*, *268*(5214), 1158.

[22]Hromádka, T; Deweese, MR; Zador, AM (2008). "Sparse representation of sounds in the unanesthetized auditory cortex". PLoS Biol. 6: e16.

[23] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, *3*(1), 79-87.

[24] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[25] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[26] LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.

[27] Li, J., Luo, W., Yang, J., & Yuan, X. (2013). Unsupervised Pretraining Encourages Moderate-Sparseness. *arXiv preprint arXiv:1312.5813*.

[28]Lin, Andrew C., et al. "Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination." Nature neuroscience 17.4 (2014): 559-568.

[29] Liwicki, M., Graves, A., Bunke, H., & Schmidhuber, J. (2007, September). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proc. 9th Int. Conf. on Document Analysis and Recognition* (Vol. 1, pp. 367-371).

[30] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML* (Vol. 30, No. 1).

[31] Machine Learning by Stanford University (Andrew Ng)

[32] Neural Networks for Machine Learning by University of Toronto (Geoffrey Hinton)

[33] Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007, June). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning* (pp. 759-766). ACM.

[34] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (No. ICS-8506). California Univ San Diego La Jolla Inst for Cognitive Science.

[35] Shang, W., Sohn, K., Almeida, D., & Lee, H. (2016). Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units. *arXiv preprint arXiv:1603.05201*.

[36] Sigtia, S., & Dixon, S. (2014). Improved music feature learning with deep neural networks. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6959-6963). IEEE.

[37] Spanne, A., & Jörntell, H. (2015). Questioning the role of sparse coding in the brain. *Trends in neurosciences*, *38*(7), 417-427.

[38] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting.*The Journal of Machine Learning Research*, *15*(1), 1929-1958.

[39] Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (pp. 1017-1024).

[40] Vinje, WE; Gallant, JL (2000). "Sparse coding and decorrelation in primary visual cortex during natural vision". *Science*. 287: 1273–1276

[41] Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., & Fergus, R. (2013). Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (pp. 1058-1066).

[42] Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

[43] Zhang, S., Bao, Y., Zhou, P., Jiang, H., & Dai, L. (2014). Improving deep neural networks for LVCSR using dropout and shrinking structure. In*Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on* (pp. 6849-6853). IEEE.